KAVEH AZADEH

Robotized Warehouses

Design and Performance Analysis

Robotized Warehouses

Design and Performance Analysis

Robotized Warehouses - Design and Performance Analysis

Gerobotiseerde magazijnen - ontwerp en prestatie-analyse

Thesis

to obtain the degree of Doctor from
Erasmus University Rotterdam
by the command of
rector magnificus

Prof.dr. F.A. van der Duijn Schouten

and in accordance with the decision of the Doctoral Board

The public defense shall be held on
Friday 26 February 2021 at 13:00 hours

by

Kaveh Azadeh
born in Tehran, Iran

Erasmus University Rotterdam

Doctoral Committee

Promoter:	Prof.dr.ir M.B.M. de Koster
Other members:	Prof.dr.ir. K. Furmans Prof.dr.ir. R. Dekker Dr.ir. N.A.H. Agatz
Co-Promoter:	Prof. D. Roy

Erasmus Research Institute of Management – ERIM

The joint research institute of the Rotterdam School of Management (RSM)
and the Erasmus School of Economics (ESE) at the Erasmus University Rotterdam
Internet: http://www.erim.eur.nl

ERIM Electronic Series Portal: http://repub.eur.nl/pub/

ERIM PhD Series in Research in Management, 515
ERIM reference number: EPS-2021-515-LIS
ISBN 978-90-5892-592-3

Design: PanArt en advies, www.panart.nl

This publication (cover and interior) is printed by Tuijtel on recycled paper, BalanceSilk®.
The ink used is produced from renewable resources and alcohol free fountain solution.
Certifications for the paper and the printing production process: Recycle, EU Ecolabel, FSC®C007225.
More info: www.tuijtel.com

FSC
www.fsc.org
MIX
Paper from responsible sources
FSC® C007225

Acknowledgments

Doing a PhD was never my plan. I figured I would just finish my Masters, get a nice job, and live a reasonably nice life. But a few random, yet significant, circumstances aligned, and I found myself at RSM, making one of the best decisions I have ever made. They say hindsight is 20:20, but my PhD is not a case of that: unlike the commonly held perception of the stressed and suffering PhD student, I truly enjoyed every moment of my PhD. And this enjoyment was only possible because of several key people who were with me throughout this journey. I would now like to pay tribute to them in this part of my dissertation.

First and foremost, I would like to express my sincerest gratitude to my advisors, René de Koster and Debjit Roy. René, thank you for your continuous support and infinite patience, and for always keeping me motivated. As Arpan rightly said, your "infectious enthusiasm" could inspire me on even the dullest days. Debjit, thank you for always being available to me, in spite of your commitments at other universities. Your boundless kindness and sharp insights significantly improved me, both as a scholar and as a person. Thank you for also hosting me in India, and for your warm hospitality! I could not have imagined having a better advisor and mentor for my PhD study.

I would like to thank Prof. Kai Furmans, Prof. Rommert Dekker, and Dr. Niels Agatz for serving on my small committee. Your constructive comments helped improve the quality of my work significantly. I am also grateful for my large committee members, Prof. Rob Zuidwijk, Prof. Yeming Gong, and Dr. Jennifer Pazour. I would like to especially thank Jennifer for all her support in the past nine years, and for planting the idea of a PhD in my mind: meeting you was one of the best things that happened to me both academically and personally. My thanks also go to Nando van Essen, Paul Haagh, and Roel Megens for their valuable insights which helped improve my work's practical relevance. I was also lucky enough to collaborate with Peter Bodnar on a research project which resulted in a fruitful and enjoyable research experience; I thank you for that.

I have spent more than six years at T9, which may raise some eyebrows for why the hell it took me so long. I guess I could have been more productive and wrap up my studies sooner, but the supportive, fun, friendly, and vibrant environment of the department provided me with minimal incentives to do so. This pleasant work environment would not have been possible without the endless efforts of Cheryl and Lianne. Thank you so much for all your work and kindness. I would like to especially thank Carmen for helping me with all my administrative work, all the beautiful plants in my office, and all the enjoyable conversations. My thanks also go to the trainers at Erasmus Sports Center, especially Garvey and Eri, for keeping me physically healthy and mentally able to enjoy food!

I am fortunate enough to be surrounded by many friends, many of whom I got to know during my time at RSM. Jelmer, thank you for being my first contact point whenever I had any problem with my research. It always amazed me how you could solve problems that I had puzzled over for hours in a matter of minutes. Jun, thank you for being such a crazy and cool office mate, and also for pushing me to an active lifestyle. Alp, thank you for taking me out of my antisocial bubble when I first joined RSM and for being such an amazing and reliable friend. Joydeep, thank you for being the subject of many of our great (!) stories, particularly the one outside McDonalds! Arpan and Jenny, thank you for introducing the concept of "Noroc" to my life. Your wedding afternoon/night/morning/the day after was one of the best experiences that I can vaguely remember. Negin, thank you for becoming one of my closest friends in such a short period of time. Sai, thank you for all the cafes, all the humor, all the ducks, and all the popeis. I would also like to thank Alberto for all the pizzas, Johann for teaching me that beer is practically water (just better), Joshua for all the badminton sessions, and Yixian for all the Chinese foods! My thanks also go to Ainara, Alex, Anna, Anirudh, Davide, Xishu, Ilona, Erik, Sabine, Nabila and many more great people I got to know at RSM. I would like to especially thank Sara and Arash for helping me feel at home from the moment I entered the Netherlands. I am also blessed to have supportive friends all around the world. In particular, I want to thank Maral, Negar, Mickal, Amin, Maryam, Haleh, and Sasan.

My deepest appreciation goes to my good friends and paranymphs, Katharina and Francesco. Frankie, thank you for being such a fantastic friend. I love our (heated) discussions about the most random things one can imagine, and it amazes me how we could manage to always be on the opposite side of any argument! Katha (or Koto, since for whatever reason you think this is your name in Farsi), thank you so much for all the Jordys, Paviljoens, THemTH, S&Cs, and Heineken zeroes. Above all, thank you for being a source of immense positivity every day, or as Carmen once said, for being the "The Sunshine" of the department during dark cloudy days (aka almost every day here in the Netherlands).

I would have never gotten to where I am today without the unconditional support of my family. Shabnam, thank you for being there for me in the most significant transition of my life. Knowing you were always around the corner calmed me down during those stressful days. Sara, thank you for being a role model of willpower for me, for always listening to my complaints, and never losing your patience with me.

Maman and Baba, thank you for teaching me to be a good person, to care for people, to be honest, and to dream big. Thank you for believing in me, even when my decisions seemed illogical. I am where I am because of your selfless support. Thank you for all the love you are giving me every day.

Finally, Maryam, I am so blessed to have you in my life. Thank you for never letting me face impossible situations alone. Your inspiring presence, encouragement, and love keep me going every single day.

Kaveh Azadeh
Rotterdam, 2020

Contents

1 Introduction

Warehouse operations tend to be labor-intensive and require large space for facilities. Large buildings are needed to store the item assortment in racks, move stock, unload and load trailers and containers, inspect picked orders, allow trucks to maneuver in the yard, and dock the trucks. Among the warehouse activities, order picking is the most laborious and expensive process. It includes collecting the right amount of the right products for a given set of customer orders. Some estimate the order picking cost to account for 55 percent of the total warehouse operating expenses (De Koster et al., 2007). Furthermore, order picking tasks are often repetitive and can suffer from poor ergonomics. Therefore, they have become the primary candidate for automation to improve efficiency in the fulfillment process.

Furthermore, unexpected major disruptions such as Brexit and the recent COVID-19 pandemic also impacted some warehouse operations. As a result of Brexit, several UK firms face difficulties finding qualified workers since they are no longer part of workers' free movement within the European Union. COVID-19 pandemic requires significant social distancing norms and new workplace protocols to ensure safety in warehouse operations. The resulting "new normal" makes it challenging to operate a warehouse with manual labor. Phase-wise automation of warehouses can be both safe and productive in the new normal times (Roy, 2020).

That being said, there is no one-size-fits-all solution for warehouse automation, and depending on the type of the warehouse and its position within the supply chain, different automated systems should be considered. In particular, retailers may choose different channels to reach their customers. For instance, they can directly ship items from the warehouse or use physical stores. Therefore, different warehouse types have emerged as a result of various distribution channels. Hence, it is crucial to understand what these channels are and how they shape different warehouse requirements.

1.1 Distribution Channels

Buying and selling goods and services electronically over the internet or e-commerce has completely changed consumers' shopping behavior. In the past, one or more visits to a brick and mortar store were required for any purchase. Today, consumers can search from a broad range of products, compare the prices of different retailers, read customer reviews about the products, and finally purchase the item by just tapping on their smartphones at 10 pm, all from the comfort of their couch. E-commerce has also provided many opportunities for businesses. Companies can extend their services beyond their geographical region and tap into the national and international markets easily. Furthermore, a retailer can market a much more diverse portfolio of products on the online platform than brick and mortar stores (Open Access Government, 2019). For instance, Walmart[1] supercenters only carry one-sixth of the number of SKUs (Stock Keeping Unit) that are carried by Walmart.com (Brynjolfsson et al., 2003). The distribution platform enabled by e-commerce that retailers use to distribute their products directly to the customers is called *Online Channel*.

The *Offline Channel*, on the other hand, is the traditional distribution platform that retailers use to distribute their products to the customers through physical stores. Despite being overshadowed by online shopping growth, physical stores still play a significant role in the consumer's shopping experience. In particular, they provide instant satisfaction from immediate possession of the purchased products (Agatz et al., 2008). Moreover, some consumers combine the two shopping experiences by browsing for the items online while making the actual purchase at the physical store, or the other way around (Skrovan, 2017; Chiou et al., 2017). That is why some of the largest e-commerce companies, such as Amazon and Alibaba, are heavily investing in having a physical presence (Schaverien, 2018; Hirnand, 2018).

Depending on how the channels are used, there are three distribution models.

- *Single-Channel*: In this model, a company only uses one of the channels to reach the customers, i.e., completely online or completely offline. Bol.com[2] and Picnic[3] are two companies that only use the online single-channel distribution model. Most of the local retail shops use an offline single-channel distribution model.
- *Multi-Channel*: Different segments of customers prefer different channels of sales and product delivery options. Therefore, retailers started to offer both online and offline channels to their customers. When retailers provide different channels to their customers that work independently of each other, it is known as a *Multi-Channel* model. Customers can purchase the products from either the physical stores or the online store. However, there is limited coordination among different channels, and the majority of the operations for each channel are done independently (Saghiri et al., 2017). For instance, each channel has a separate warehouse. Even if they use the same warehouse facility for both channels, most warehousing operations such as storage, picking, packing, and shipping of the orders are entirely separated for each channel. Albert

[1] Walmart is an American multinational retail corporation that operates a chain of hypermarkets, discount department stores, and grocery stores
[2] Bol.com is the leading webshop in the Netherlands for books, toys, and electronics
[3] Picnic is an online supermarket in the Netherlands

Heijn[4] and Jumbo[5] are examples of companies deploying a multi-channel distribution model (Dijkhuizen, 2020; De Weerd, 2019).

- *Omni-Channel*: In this model, retailers also use both channels to reach customers. However, unlike the multi-channel model, all channels are integrated seamlessly with each other. Customers can buy their products from any channel, receive it in any of the available delivery options, and, if required, return the product via any available medium. For example, retailers offer purchasing options such as Buy-Online-Pickup-In-Store (BOPIS, or click-and-collect), Buy-Online-Return-In-Store (BORIS) and delivery options such as ship product from one store to another store, and locker pick-up (Uichanco et al., 2019). De Bijenkorf and Blokker[6] are examples of companies that use an omni-channel model (Dijkhuizen, 2019a,b).

Figure 1.1 illustrates these three distribution models. The dashed arrows correspond to the physical goods flow, and the solid arrows correspond to the flow of information. Next, we discuss the different types of warehouses that have emerged due to various distribution channels.

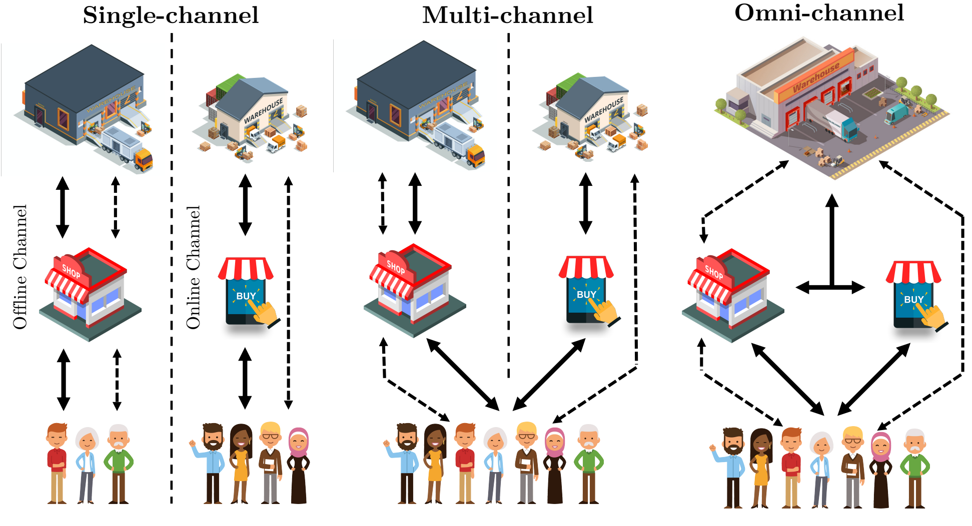


Figure 1.1: Distribution channels (⇢ physical goods flow, → information flow)

[4] Albert Heijn is the largest and most famous Dutch supermarket chain with more than 1000 stores in the Netherlands and Belgium
[5] Jumbo is the second largest Dutch supermarket chain with more than 600 stores in the Netherlands
[6] Blokker is a Dutch homeware retailer

1.2 Different Warehouse Types

In an offline channel, warehouses act as a distribution center for store replenishment. We call these warehouses *Store Replenishment Warehouses*. With the start of e-commerce, store-based retailers started to transform their warehouses to incorporate the online channel. In an online channel, although a small number of customer orders could be fulfilled from stores, in large-scale operations, the orders are typically fulfilled directly from a warehouse. In the beginning, when the share of e-commerce was still relatively small, a small part of the store replenishment warehouse was dedicated to serving online orders as an ad-hoc solution. However, with the online channel's significant growth, fulfilling all orders from the same facility became difficult. Particularly, three main reasons forced retailers to start a dedicated *E-commerce Warehouse*:

1. *Order Profile*: Traditional store-based retail warehouses are accustomed to daily store replenishment with a large number of daily order lines and large volumes per order line. In contrast, in online retail, the number of daily orders can be much larger, with only a few lines per order. For instance, the average order size at Amazon warehouses in Germany is 1.6 items per order (Boysen et al., 2019). Moreover, store replenishment orders often consist of pallets or overpacks, whereas online customer orders are for piece quantities. Hence, handling online orders requires a different approach than the store replenishment orders.

2. *Storage Space*: The storage cost in a warehouse is much lower compared to a store shelf. Therefore, online retailers can afford to offer a much larger assortment of products on their webshop since they do not have the physical store's cost and space limitation (Brynjolfsson et al., 2003). Therefore, with the strong growth of e-commerce, the small part of the store-based retail warehouse that was initially dedicated to the online channel could no longer accommodate the large product assortment.

3. *Fast Delivery*: Traditional store-based retail warehouses were not designed for fast delivery. They aimed to replenish stores in time to prevent stock-outs. Consequently, most warehouses were located in relatively remote but strategic locations from which stores could be replenished with an acceptable lead time. When the online channel's share was small, it was still possible to accommodate occasional fast deliveries for online customers from the same store-replenishment warehouse. But with the growth of online customers and rising expectations of quick deliveries, e.g., next-day or even same-day delivery, it was not possible to meet customer demands from the same warehouse.

Many retailers have continued to operate with separate single-channel warehouses: store replenishment warehouse, usually located in remote areas, and e-commerce warehouses, usually located near urban areas (see Figure 1.2). Especially, retailers with a large volume of online daily orders do not have any other option but to operate with separate warehouses. Meanwhile, several retailers with a moderately large number of daily online orders have combined the two operations in *Omni-Channel Warehouses* (see Figure 1.3). The high cost of land, especially in regions like Western Europe, and the shortage of labor for warehouse work, have made it difficult for many retailers to maintain multiple warehouses. Furthermore, operating separate warehouses results in duplication of inventory. In contrast, in an omni-channel warehouse, online and offline orders are fulfilled from the same inventory thanks to new technological advancements that allow handling online customer orders and store replenishment orders simultaneously from the same facility. De Bijenkorf and Blokker are examples of companies that have recently merged the warehouse operations in a single omni-channel warehouse (Dijkhuizen, 2019a,b).

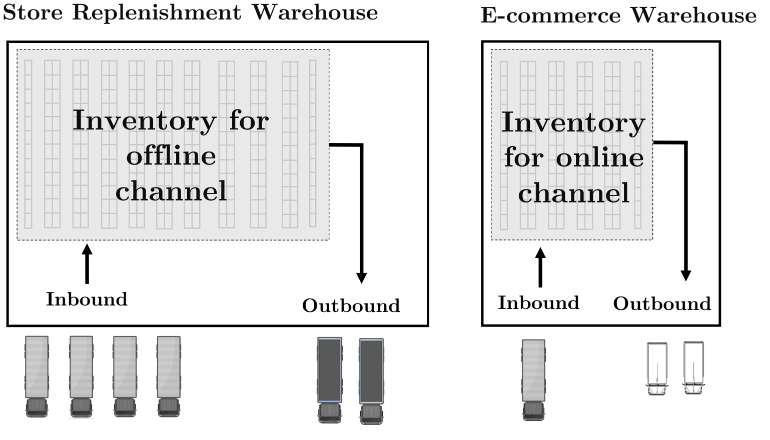


Figure 1.2: Inventory and goods movement in single-channel warehouses

1.3 Warehouse Automation

We identify three warehouse types depending on the distribution model, each with different characteristics and requirements. Therefore, the choice of a suitable automated system is different depending on the warehouse type.

Automation for Store Replenishment Warehouses: The main objective of a store replenishment warehouse is to replenish stores at the due time to avoid stock-outs. These warehouses should fulfill orders with many lines with large volume per line, i.e., pallets or overpacks, from a medium assortment of products under a moderate time pressure (Boysen et al., 2020). Furthermore, they do not require much throughput flexibility since the stores' demand pattern is more or less fixed with predictable peaks (Kembro et al., 2018). In normal circumstances, store replenishment warehouses have weekly cycles, often with peaks

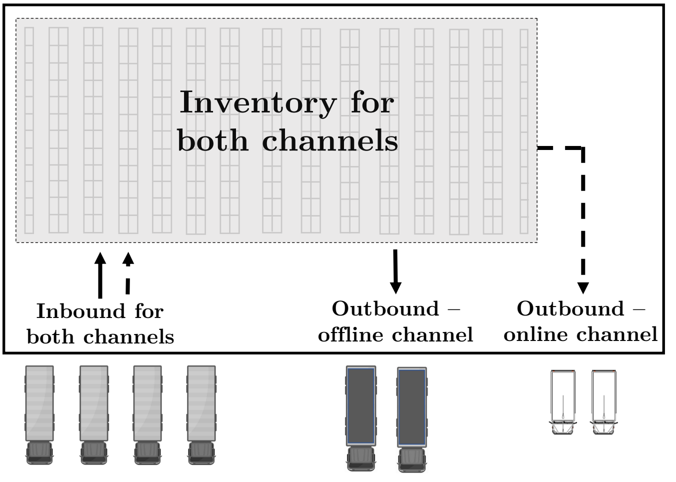


Figure 1.3: Inventory and goods flow in an omni-channel warehouse (- - → online channel goods flow, → offline channel goods flow)

on Fridays, to ensure stores are replenished for the peak sales in the weekends, and on Mondays, to refill the stores afterward (Boysen et al., 2020). Therefore, an automated solution should be able to handle pallets and large box items with moderate throughput flexibility. *Fully-Automated Case Picking* is an example of a fully automated order fulfilment process for store replenishment warehouses. In this system, incoming goods, predominantly homogeneous unit load pallets, are first stored in an Automated Storage and Retrieval System (AS/RS). When a certain product is requested, the pallet is retrieved and moved into a depalletizing stage. In this step, pallets are broken down into individual cases by an industrial robot. The loose cases are then transported with a conveyor to be stored using a mini-load AS/RS. Once a store places an order, the cases are retrieved from the AS/RS and transported to a palletizing stage. There, another industrial robot stacks the loose cases to create mixed pallets or roll cages according to the store order (Boysen et al., 2020).

Automation for E-commerce Warehouses: E-commerce warehouses should fulfill small-sized orders, from a large assortment of products, under significant time pressure, and need to be flexible enough to adapt to unpredictable demand fluctuations (Boysen et al., 2019). Traditional goods-to-men automated systems, such as mini-load AS/RS, are expensive and inflexible with a long implementation time, making them less suitable for an e-commerce warehouse. These issues have given birth to robot-based picking solutions. These systems use free-roaming retrieval robots, such as shuttles, free-roaming Autonomous Guided Vehicle (AGVs) and Autonomous Mobile Robot (AMRs), to improve picking efficiency. Although they are a bit slower in terms of hourly pick rate than conventional automated systems such as mini-load AS/RS, they are preferred due to their lower cost, quick deployment, flexibility, and scalability. Shuttle-based storage and retrieval systems and robotic mobile fulfilment systems are two examples of such robotic solutions for e-commerce warehouses.

Automation for Omni-channel Warehouses: The main challenge in an omni-channel warehouse is the presence of small-sized customer orders along with large-sized store replenishment orders. Therefore, the automated solution should be able to pick orders with few and with many order lines. Robotic solutions, in particular AMRs, can pick for various order sizes, which make them a viable candidate for order picking in an omni-channel warehouse. Pick-support AMRs (PS-AMR) are an example of such a robotic solution.

Warehouse automation requires considerable scale and a long-term vision, as the investments can be earned back only in the medium and longer-term. Therefore, it is crucial to develop tools to help decision-makers find the correct solutions for their warehouses. In this thesis, we aim to provide useful academic and practical insights by modeling and optimizing the performance of different automated and robotic picking systems.

1.3.1 Research Opportunities

The majority of warehouse research still focuses on conventional storage and order picking methods. Due to rapid system developments, it is time for an update, as the new technologies have provided new and interesting research opportunities. Therefore, in Chapter 2, we structure the latest automated technologies and give an overview of these technologies and the research. We also review the modeling techniques used and the research opportunities they provide. In this chapter, we do not limit ourselves to a particular warehouse type and review systems that are used in all three warehouse types.

In Chapter 3, we turn our attention to e-commerce warehouses. The main challenge in many fulfillment centers is to adapt the picking capacity to the order volume required. This is more pronounced in e-commerce rather than store replenishment warehouses due to unpredictable demand fluctuations. The Shuttle- or Autonomous Vehicle-based Storage and Retrieval System (AVS/RS) is one very popular candidate to address this challenge. In this system, a combination of autonomous shuttles and lifts are used to perform the order fulfillment process. In each tier, shuttles move autonomously in the horizontal directions using rails and are transported in the vertical direction between tiers using lifts. We categorize these systems as *Horizontal* systems (see Figure 1.4).

The major problem with these systems is that the system throughput is constrained by the number of lifts present in the system, limiting their flexibility to react to a change in demand. Recently, robotics-based storage and retrieval systems have been developed to address this issue by eliminating the multi-touch retrieval process of AVS/R systems. In these systems, a single robot can move independently and autonomously in the horizontal and vertical directions inside the rack structure to transport items between storage locations and

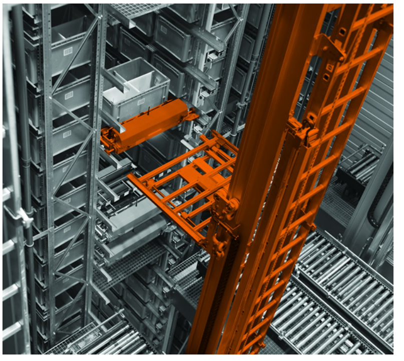
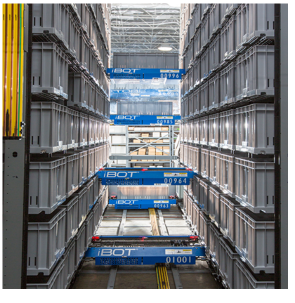
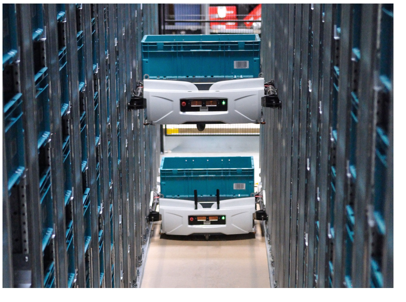


Figure 1.4: Horizontal system (Source: Vanderlande)

workstations. Therefore, we categorized these systems as *Vertical* systems (see Figure 1.5). Many studies exist that describe and analyze the horizontal systems' performance, while the vertical system has not been studied yet. Furthermore, there are fundamental differences between the two systems, which leads to a different modeling approach and different layout designs and control policies for the vertical system. Therefore, in Chapter 3, we first investigate the vertical system in more detail, and then we compare its performance and costs with the horizontal systems.



(a) PerfectPick (Source: OPEX)



(b) Skypod$^{\mathrm{TM}}$ (Source: EXOTEC)

Figure 1.5: Vertical system

In Chapter 4, we study a system that can be used in all three warehouse types. In this system, PS-AMRs collaborate with human pickers to carry out the order fulfillment (see Figure 1.6). In this collaborative environment, the picker accompanies the AMR only for item picking, and the AMR autonomously carries out the remaining travel and drop off functions. Manual pickers and pickers collaborating with PS-AMRs can work side-by-side, making this collaborative system ideal for companies who want to automate their manual system but are skeptical about the investment costs. Companies can start with a small number of PS-AMRs and gradually expand this over time without affecting their current pick process, reducing investment cost and automation risk significantly. The parallel movement of pickers and AMRs makes the modeling, analysis, and optimization of this system completely different from fully manual picking systems or other robotic systems. Therefore, we dedicate Chapter 4 to a detailed analysis of such systems. Particularly we investigate optimal operational policies when using PS-AMRs in an omni-channel warehouse.

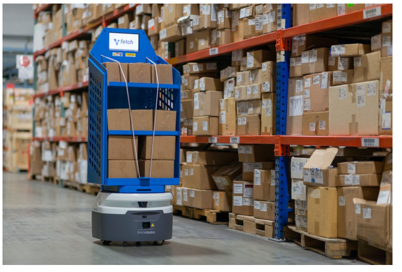


Figure 1.6: Pick-Support AMR (Source: Fetch Robotics)

1.4 Contribution and Thesis Outline

Chapter 2: Robotized and Automated Warehouse Systems: Review and Recent Developments[7]

This chapter reviews new categories of automated and robotic handling systems, such as shuttle-based storage and retrieval systems, shuttle-based compact storage systems, and robotic mobile fulfillment systems. Particularly, we aim to answer the following research questions:

- What is the current state-of-the-art academic literature focusing on automated and robotic handling systems?
- What are the key research methods deployed to analyze the performance of these systems?
- What are the prime areas for further academic research?

For each system, we categorize the literature into three groups: system analysis, design optimization, and operations planning and control. Our focus is to identify the research issue and operations research modeling methodology adopted to analyze the problem. We find that many new robotic systems and applications have hardly been studied in academic literature, despite their increasing use in practice. Because of unique system features (such as autonomous control, flexible layout, networked and dynamic operation), new models and

[7] Azadeh, K., De Koster, R., and Roy, D. (2019). Robotized and automated warehouse systems: Review and recent developments. *Transportation Science*, 53(4):917-945.

methods are needed to address the design and operational control challenges for such systems, particularly for the integration of subsystems. Integrated robotic warehouse systems will form the next category of warehouses. All vital warehouse design, planning, and control logic, such as methods to design layout, storage and order-picking system selection, storage slotting, order batching, picker routing, and picker to order assignment, will have to be revisited for new robotized warehouses.

Chapter 3: Design, Modeling, and Analysis of Vertical Robotic Storage and Retrieval Systems [8]

This chapter builds a framework to analyze the performance of the vertical system and compare its throughput capacity with the horizontal system. We aim to answer the following research questions:

- How do we build accurate and efficient analytical models to analyze the performance of the vertical system?
- What is an optimal layout for the vertical system in terms of throughput performance?
- How do the blocking delays affect the throughput performance of the vertical system?
- Which system is better in terms of costs and throughput capacity: horizontal or vertical?

We build closed queuing network models to estimate the throughput performance of the system. The performance measures are, in turn, used to identify the optimal system design parameters. The results show that the optimal height-to-width ratio in time of a vertical system is around one. Because a large number of system robots may lead to blocking and delays, we compare the effects of different robot blocking protocols on the system throughput: Robot Recirculation (REC) and Wait-on-Spot (WOS). The WOS policy produces a higher system throughput when the number of robots in the system is small. However, for a large number of robots in the system, the REC policy dominates the WOS policy. Finally, we compare the operational costs of the vertical and horizontal transport systems. For systems with one load/unload (L/U) point, the vertical system always produces a similar or higher system throughput with a lower operating cost compared with the horizontal system with a discrete lift. It also outperforms the horizontal system with a continuous lift in systems with two L/U points.

Chapter 4: Dynamic Human-Robot Collaborative Picking Strategies [9]

One popular way of warehouse automation is with Autonomous Mobile Robots (AMRs) that collaborate with human pickers to efficiently pick the orders by reducing the pickers'

[8] Azadeh, K., Roy, D. and De Koster, R., (2019). Design, modeling, and analysis of vertical robotic storage and retrieval systems. *Transportation Science*, 53(5):1213-1234.

[9] Azadeh, K., Roy, D., and De Koster, R. (2020). Dynamic human-robot collaborative picking strategies. *Available at SSRN*

unproductive walking time. Picker travel time can be reduced even more by zoning the storage area. In this strategy, the warehouse is divided into multiple storage zones, with one or multiple pickers assigned to each zone. Pickers only pick from their dedicated zones. In every zone, the robot is paired with a picker from that zone, and together they pick all the required pick list items from that zone. If the order is incomplete, the robot progresses to another zone. Else, if all needed items are picked, it travels back to the depot, and the picker becomes available for processing the next order. We call this picking strategy a *Progressive Zoning* (PZ) strategy. There is also a *No Zoning* (NZ) strategy in which the robot is paired with any available picker, and together they pick all the pick list items from the whole warehouse. Few zones are particularly good for the large store replenishment orders, while many zones are particularly good for the small online orders. However, the optimal zoning strategy for an omni-channel warehouse using these robotic systems is not clear since they usually process various order sizes. In this chapter, we study the effect of dynamic zoning strategies, i.e., dynamic switching between NZ strategy and PZ strategy. We aim to answer the following research question:

- Is it possible to achieve a higher pick performance with lower operational costs in a human-robot collaborative picking system by dynamically switching between the pick strategies, given a fixed number of resources?

We solve the problem in two stages. First, we develop queuing network models to obtain load-dependent pick throughput rates corresponding to a given number of AMRs and a picking strategy with a fixed number of zones. Then, we develop a Markov decision model to investigate how higher pick performance can be achieved by dynamically switching between these pick strategies. Using data from an omni-channel warehouse that processes orders of various sizes, we show that a Dynamic Switching (DS) policy can lower operational costs by up to 7 percent. However, these cost savings decrease as the number of robots per picker increases.

Research Statement

This Ph.D. thesis has been written during the author's work at the Erasmus University Rotterdam. The author is solely responsible for formulating the research questions, building the analytical models, analyzing the results, and writing all the chapters of this thesis. While carrying out the research, the author received valuable and constructive feedback from the doctoral advisors and other doctoral committee members, which subsequently increased the quality of research. Chapters 2 and 3 are published, and Chapter 4 has been submitted to a scientific journal and is undergoing the review process.

2 Robotized and Automated Warehouse Systems: Review and Recent Developments

2.1 Introduction

Warehouse operations tend to be labor intensive and require large space for facilities. Large buildings are needed to store the item assortment in racks, to move stock, to unload and load trailers and containers, to inspect picked orders, to allow trucks to maneuver in the yard, and to dock the trucks. With the advent of e-commerce, companies store millions of unique items and handle large and variable daily order volumes. On the other hand, the most laborious and expensive process, order picking, is repetitive, often suffers from poor ergonomics, and requires high-quality labor willing to work in shifts, which is often difficult to get. It is therefore not surprising that warehousing systems and processes are key candidates for automation. In addition, the land available for warehouses (which should preferably be close to the demand points) has become scarce, and many warehouses have to operate 24/7. Together, this has given warehouse automation a big boost.

Warehouse automation dates back to the 1960s, when the first high-bay (20-40 m high was quite standard) unit-load warehouses were established in Germany with aisle-captive cranes driving on rails, constructed as a silo building (Industrie-forum, 2004). These so-called AS/R (automated storage and retrieval) systems were able to store bulk stock on unit loads (pallets, or totes: miniload system). They could also work in conjunction with manual pick stations as a parts-to-picker system, where the retrieved unit load was restored after picking units from it.

Since then, AS/R systems have become very popular in practice, and research has gained momentum with the papers by Hausman et al. (1976), and Bozer & White (1984). Hundreds of papers have been published on these systems. An overview on AS/R systems classification and research studies is given by Roodbergen & Vis (2009).

During the last decade, warehouse automation has developed rapidly. A big boost has been given by the AVS/R (autonomous vehicle-based or shuttle-based storage and retrieval) systems. These systems use racks with aisles and deploy autonomous shuttles that operate at each level in each aisle. Vertical transport is enabled by lifts. Another important development has been automated pallet stacking and destacking technologies, in particular also by mixed-case palletizing technology developed in the early 2000s. A new generation of Autonomous Mobile Robots (AMRs), supporting the order picking process has recently been introduced. These systems will gradually result in automated picking processes. Pioneered by Witron, combining multiple technologies has led to the advent of completely automated warehouses, particularly in the store-based retail industry (mostly grocery). Based on the authors' experience, in Western Europe alone, about 40 fully automated warehouses are in operation and many are under development. Although these warehouses are large, they are much smaller (and supposedly more cost-efficient) than their conventional, manual counterparts. Figure 2.1 shows a flow diagram of such a warehouse with typical storage and handling systems.

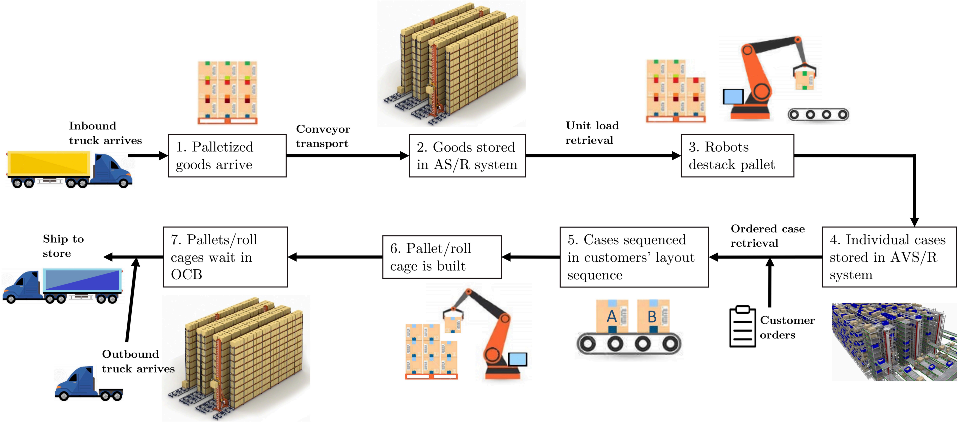


Figure 2.1: Material flow in a typical automated warehouse

In such an automated retail warehouse, selected suppliers unload their own trucks and feed the pre-announced single-SKU (stock-keeping unit) pallets to a check-in conveyor (step 1). The pallets are then stored in an AS/R system (2). When a certain product is requested, the pallet is off loaded and automatically destacked (3). The loose cases are then often put on trays to ease manipulation and are stored in a miniload AS/R, or in an AVS/R system (4). When the store order arrives, the cases are retrieved and sequenced (5), and mixed-case palletizers build the pallets or roll-cages in a store-specific sequence that allows rapid shelving in the store (6). These roll-cages then wait in an order consolidation buffer (OCB), usually an AS/R system (7), until the departure truck arrives, after which they are retrieved and loaded in the sequence determined by the stop sequence in the truck route.

Apart from the (many) technicians needed to keep the system alive, no manual handling is involved. In addition to these fully automated warehouses, many partially robotized warehouses have been built. According to Buck Consultants International (2017), in the

Netherlands alone 63 large new warehouses were constructed in the period 2012-2016, using robot technologies. However, the majority of warehouse research still focuses on conventional storage and order picking methods. The overview by De Koster et al. (2007) provides some avenues for research into (semi-)automated picking methods. Due to rapid system developments, it is time for an update, as the new technologies have provided new and interesting research opportunities. This paper structures the new automated technologies and provides an overview of these technologies and the research carried out already. It also reviews the modeling techniques used and the research opportunities they provide. We focus on the design and control aspects of order picking systems because they form the heart and soul of any warehouse. In doing so, we include the corresponding automated product storage and handling techniques. Figure 2.2 categorizes the automated picking systems, both the classical as well as the newly developed automated picking systems.

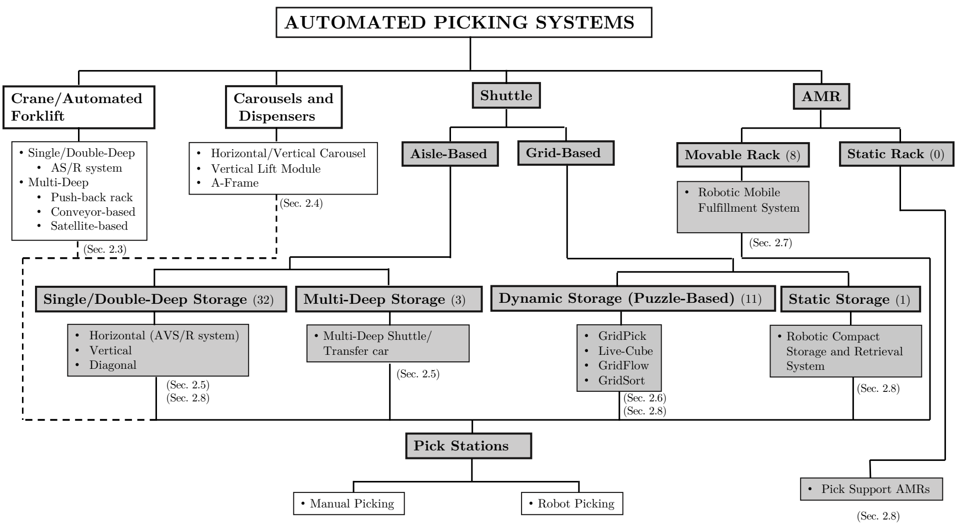


Figure 2.2: Classification of automated picking systems. The literature of the gray-shaded systems is reviewed. The numbers placed next to the systems indicates the number of reviewed papers.

In this study, we focus on recent robotic automated picking systems, in particular systems that use free-roaming retrieval robots such as shuttles and free-roaming AMRs (the grey shaded systems in Figure 2.2). The more conventional systems, such as cranes, automated forklifts, carousels and automated dispensers have been reviewed in other papers (Roodbergen & Vis, 2009; Litvak & Vlasiou, 2010; Gagliardi et al., 2012; Boysen & Stephan, 2016); we only highlight a few key articles. To find articles, we used the following search terms in Scopus: “autonomous vehicle/shuttle storage and retrieval systems”, “robotic mobile fulfillment system", "puzzle-based storage system", "compact warehouse storage systems" and "robotic warehouse storage and retrieval systems", as well as variants of these search terms. We review papers published in high quality journals, complemented by some working papers and proceedings for prominent systems that have not received much attention yet. We review 54 papers on the core systems indicated in the gray-shaded boxes in Figure 2.2.

We first describe various modeling methods used in the design and operation of the systems and the associated objectives (Section 2.2). Section 2.3 deals with the 'conventional' AS/R systems, that have been researched intensively, and then continues with less conventional crane and automated forklift-based systems, such as multi-deep racks operated by cranes and satellites. Section 2.4 discusses different types of carousels, Vertical Lift Modules (VLM), and automated dispenser systems. Section 2.5 discusses various types of aisle-based AVS/R systems, and Section 2.6 considers grid-based storage and retrieval systems. Section 2.7 continues with robotic movable rack-systems. Section 2.8 discusses directions for future research and includes emerging technologies, in particular, humans picking in collaboration with AMRs. We conclude in Section 2.9.

2.2 Modeling Methods and Objectives in Storage, Transport and Order Picking Process

Two approaches exist to model the systems: *Analytical*-based and *Simulation*-based. Simulation-based models can mimic reality accurately and produce the least error. However, conceptualizing and designing a detailed and accurate simulation model is time intensive. Optimizing the entire design space may require the development of multiple models. Therefore, at an early stage, analytical models are preferred, to reduce the design search space and to identify a limited number of promising configurations. Compared to simulation modeling, analytical models run faster and can obtain the optimal configuration either directly or with a quick enumeration over a large number of design parameters. The error made in the estimated performance measures using analytical models is usually acceptable for the conceptualization phase. Section 2.2.1, explains analytical models. Section 2.2.2 discusses what the different objectives and decisions are in evaluating automated warehouses and how the analytical models are used to optimize those objectives. We also present the classification scheme that we use for reviewing articles.

2.2.1 Analytical Models

The most common analytical models for storage and retrieval are classified into three categories: *Linear and Mixed-Integer Programming Models*, *Travel Time Models*, and *Queuing Network (QN) Models*.

Linear and Mixed-Integer Programming Models

Many of the design and operational decisions in automated systems can be optimized using Linear Programming (LP) or non-linear and Mixed Integer Programming (MIP) models. For instance, LP and MIP models can be used for optimizing the shape of the system, obtaining the right choice of storage policy, scheduling and sequencing order transactions, and establishing order batching rules. LP and MIP models are usually used in a deterministic setting. To capture the stochasticity, travel time and queuing network models are preferred.

Solution Methods for Linear and Mixed-Integer Programming Models: LP models can be solved exactly in polynomial time. However, the exact solutions for the majority of the MIP models are intractable. As a result, metaheuristic algorithms are developed which provide near optimal solutions in a short time. The notion behind metaheuristic algorithms is to find the best solution out of all possible feasible solutions. Some notable example of metaheuristic algorithms include genetic algorithms, tabu search, simulated anealing, and adaptive large neighborhood search. See Glover & Kochenberger (2006) for a more detailed overview of the different metaheuristic algorithms. Recent developments in exact and heuristic algorithms have resulted in an integrated technique called matheuristics. In this method, the problem is decomposed into several small sub-problems which can be solved using exact algorithms. Later, the results of sub-problems are used in the heuristic algorithm (see Puchinger & Raidl (2005)).

Travel Time Models

Using travel time models, the design engineer can obtain the amount of time that it takes for a resource to move from one location to another. For instance, in an automated parts-to-picker picking context, travel time models can be used to obtain a closed-form expression for the expected load storage and retrieval time. The closed-form travel time expressions are usually simple and computationally friendly. Therefore, they can be used to limit the search space before adopting a detailed simulation, or for optimizing the design choices. They can also be used to estimate the expected service time of a server in a network of queues. Despite the simplicity of the travel time models, they are not capable of capturing several factors such as interaction between multiple resources, parallel processing by multiple resources, or queuing within the system. In these scenarios, QN models are preferred.

Queuing Network Models

Automated picking systems can be modeled as a multi-stage service system using a QN. In a QN, a customer arrives in the system, undergoes several stages of service and leaves the system. Several types of queuing networks have been studied: *Open* (OQN), *Closed*

(CQN), and *Semi-Open* (SOQN). In an OQN, customers, such as orders to be picked, arrive from an external source and after receiving service in different nodes, they leave the system. An OQN is particularly useful to estimate expected order throughput time. However, in many systems, resources accompany orders during the whole or a part of the process, e.g., a transport vehicle, or a transport roll container or a pallet. Often, the number and the capacity of the resources are limited that affect the performance of the system. For instance, orders might be transported by expensive robots in the system. In this scenario, an OQN is not capable of accurately estimating the performance of the system as it assumes an infinite supply of robots. One way to overcome this challenge is to model the system as a CQN. In a CQN, a limited number of resources are paired with the incoming orders. Once an order is completed, the resource becomes available to serve another order. The limited number of resources enforces a population constraint in the CQN. However, it is implicitly assumed that an infinite number of orders are waiting outside the system (Heragu et al., 2011). CQNs are useful to estimate the maximum throughput capacity of the system. Using a CQN to model the systems in which the incoming customers and the resources are paired together throughout the process, leads to an underestimation of the true customer waiting time. The reason lies in the assumption (infinite number of customers waiting externally in a CQN). However, in reality, there are times when a customer needs to wait for a resource or vice versa. In this situation, an SOQN is a suitable model because it can accurately capture the external transaction waiting time. As it illustrated in Figure 2.3, an SOQN (in the literature sometimes called an open queuing network with limited capacity) possesses a synchronization station in which incoming customers waiting at an external queue are paired with available resources in the resource queue. Then, the customer is processed using the resource that carries the customer to pre-specified different nodes (Cai et al., 2013; Roy et al., 2015b; Roy, 2016).

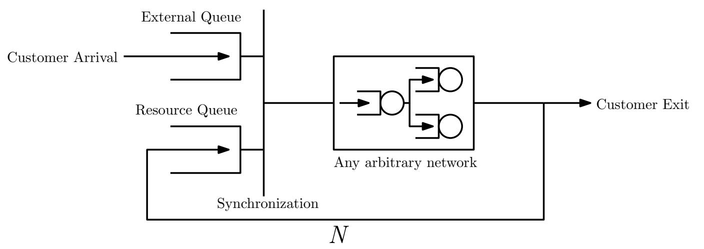


Figure 2.3: A general semi-open queuing network with N circulating resources

Solution Methods for Evaluating Queuing Networks: One of the most important methods for calculating performance measures of product-form queuing networks (Baskett et al., 1975) is *Mean Value Analysis* (MVA) (Reiser & Lavenberg, 1980). The MVA algorithm is based on Little's Law and the arrival theorem. However, networks used in analyzing automated picking systems usually do not have product-form solutions for a number of reasons, such as non-exponentially distributed service times, customer blocking, or non-Markov routing. Therefore, approximation algorithms are used to estimate the performance measures of the system. Several approximation techniques such as Approximate Mean Value Analysis (AMVA) and the parametric decomposition approach proposed by Whitt (1983) have been developed based on the characteristics of the network. Bolch et al. (2006) provide a detailed overview of exact and approximate algorithms to evaluate the performance of open and closed queuing networks. The SOQN does not have a product-form solution, even for Poisson arrivals and exponential servers. The Matrix-geometric method (MGM), aggregation, network decomposition, parametric decomposition, and performance bounds are the most common solution approaches for approximating the performance of an SOQN. A detailed overview of solution techniques to evaluate an SOQN is presented in Jia & Heragu (2009) and Roy (2016). When it is not possible to analytically solve a queuing network, it is always possible to obtain its performance measures by simulation.

2.2.2 Decision Variables and Performance Objectives

Two levels of decision-making can be distinguished in warehouse planing and design: long-term (tactical) and short-term (operational).

In long-term planning, decisions revolve around the hardware design selection and optimization (DO) of the system. At this level, the prime objective is to maximize the throughput and the storage capacity of the system. The objectives are affected by several decision variables, such as the physical layout configuration (e.g., the number of aisles, the depth of each aisle, the number of cross-aisles, and the number of tiers), the number of robots and lifts, and the number and location of load/unload points and workstations. At this stage, the focus is on the decisions that are hard to alter once the system is in place.

Short-term decision-making focuses on operational planning and control (OP&C). The prime objectives are to minimize lead time, waiting time, response time, and resource idleness, etc. Decisions include vehicle assignment policies, blocking prevention protocols, dwell point use of the vehicles, i.e., selecting the location where a vehicle without a job (idle vehicle) is parked, storage slotting, and workstation assignment rules.

Analytical models can address both the long-term and short-term decision-making. LP models are used to optimize any objective function (e.g., cost) while satisfying multiple constraints. With a (usually non-linear) travel time model, it is (sometimes) possible to obtain a closed-form expression of the performance measures, such as the average processing time. By taking derivatives with respect to the desired decision variables, one can optimize the system with regards to the performance measure. However, deriving a closed-form expression of system measures such as transaction time (including waiting) is often not possible. For this purpose, queuing network and simulation-based models are used. Design performance optimization then is done by enumerating the decision variables. Sometimes, combinations of decision variables have a joint effect on the performance of the system. As a result, some authors, such as Ekren & Heragu (2010b) suggest using regression models with interaction variables to evaluate the combined effect of decision variables on the performance of the system. Then, the enumeration is done over the variables and their combinations to examine the effect on the desired performance measure.

Table 2.1 presents a framework of different objectives and decision variables and the suitable modeling approach to address them.

Table 2.1: Decision-making framework and appropriate modeling methods

Decision Level	Prime Objectives	Decision Variables	Modeling Approach
Long-Term Decisions	Maximize:	Physical layout:	Simulation
(Design Optimization)	Throughput capacity	number of aisles	Travel Time Model
	Storage capacity	number of cross aisles	Closed Queuing Network
		depth of the aisle	Semi-Open Queuing Network
		number of tiers	Deterministic Optimization (LP,IP,MIP)
		Number of robots	
		Number of lifts	
		L/U point(s)	
		workstation(s) location	
Short-Term Decisions	Minimize:	Vehicle assignment policy	Simulation
(Operational Planning	Lead time	Block prevention policy	Travel Time Model
and Control)	Waiting time	Dwell point policy	Closed Queuing Network
	Response time	Storage policy	Semi-Open Queuing Network
	Resource idleness	Resource scheduling	Deterministic Optimization (LP,IP,MIP)
		Sequencing transactions	

When reviewing the articles in Section 2.5, Section 2.6 and Section 2.7, we leverage the presented framework in Table 2.1 and group the articles based on the prime objective being investigated. The categories include: System Analysis, Design Optimization, and Operations Planning and Control. System analysis articles focus on modeling techniques to estimate the performance of the system without focusing on any optimization. Design optimization articles focus on hardware optimization of the system (e.g., system layout), and operations planning and control articles focus on the software optimization of the system (e.g., block prevention policies).

2.3 Automated Storage and Retrieval Systems with Cranes or Automated Forklifts

Crane-based Automated Storage and Retrieval Systems (AS/RS) were introduced in the 1960s. Initially, their main application was in pallet warehouses storing bulk inventories. Later, mini-load warehouses and more compact multi-deep order picking warehouses were also automated. In this section, we discuss the different types of crane/automated forklift-based automated storage and retrieval systems, as mentioned in Figure 2.2.

2.3.1 Single/Double-Deep Storage

Such a system consists of racks and automated handling systems such as cranes or automated forklifts. These handling systems can be aisle-captive (typically cranes) or aisle-roaming (typically high-bay automated forklifts). To perform a storage operation, a crane picks up a load, usually from a conveyor, and stores it in the 30-40m high racks. Driving and lifting in the aisle take place simultaneously. The process sequence is reversed for a retrieval operation. It is also possible to carry out a dual command cycle, in which a storage and a retrieval job are combined. This would save one movement per dual command cycle; however, there may be an additional wait for pairing a storage transaction with a retrieval. If totes instead of pallets are stored, the system is referred to as mini-load. Figure 2.4 shows an example of such a warehouse.

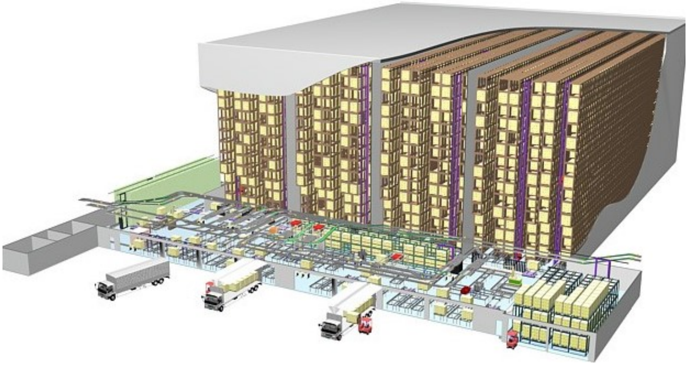


Figure 2.4: Automated high-bay warehouse for pallets with aisle-captive cranes (De Koster, 2015)

Unit-load and mini-load aisle-captive single-deep AS/R systems have been studied extensively. One of the first scientific articles is by Bozer & White (1984). They calculate the average cycle time of the crane for single command cycles, and assume that crane travel to any location within the rack has the same probability (random storage policy). Their expected cycle time is $E[T] = \left(1 + \frac{(t_y/t_x)^2}{3}\right).t_x$, in which t_x is the travel time to the farthest location in the rack and t_y is the lifting time to the highest location in the rack. The formula assumes that the crane drives and lifts at the same time and that the travel time to the farthest location is longer than the lifting time. Using this formula, the optimal ratio between the length and height of an aisle can be obtained, which proves to be square in time (SIT), meaning that the travel time to the farthest location and the lifting time to the highest location are identical. Assuming that a crane travels approximately four times faster than it lifts, the length of the aisle should therefore be four times its height in order to minimize the cycle time. Later on, this formula was adjusted to include other aspects of the warehouse, such as different storage strategies (such as ABC storage), dual command cycles, and different locations of the load and unload points (the above formula assumes one such point, at the lower corner of the rack). We refer to Roodbergen & Vis (2009) for an extensive overview of the literature on AS/R systems. Furthermore, Gagliardi et al. (2012) provide an overview of the simulation-based models for AS/R systems. Boysen & Stephan (2016) present a novel classification schemes for defining various crane scheduling problems in AS/R systems. Later, they applied the scheme to review the literature.

In the case of ABC (or product turnover-based) storage, the items are divided into classes (e.g., three: A, B, C), based on item turnover rate. The locations are also divided into groups based on travel time to the L/U point. This ensures that the items from the class with the highest turnover rate are located closest to that point. Hausman et al. (1976) investigated the cycle time calculations with ABC storage and EOQ-based replenishment. Later, their results were extended to N product classes by Rosenblatt & Eynan (1989). Hausman et al. (1976) calculated the optimal class boundaries for known ABC demand curves, for example, 20/70 demand curves, whereby 20% of the items (or unit-loads) are responsible for 70% of the demand. In the calculation, they considered product restocking according to a continuous review <s, Q> policy, with the stocking quantity Q being equal to the optimal order quantity. However, they did not take into account that the more storage classes there are, the fewer items are stored per class. This requires more space per item stored in the class, since the space within the classes cannot be shared by the items which lengthens crane travel time. In the extreme case of one item per class, the space required is $\sum \lceil Q_i + SS_i \rceil$ whereas in the extreme case of one class containing all items (i.e., random storage), the space required is $\sum \lceil \frac{Q_i}{2} + SS_i \rceil$. This means that an optimum number of storage classes can be distinguished. In practice, the optimal number of classes is small (about 3 to 5,) but the cycle time is relatively insensitive to the exact number. At such a limited number of classes, products can perfectly share the space available in the class. However, the required number of locations on top of the average stock level quickly amounts to an additional 40% (Yu et al., 2015).

2.3.2 Multi-Deep (Compact) Storage

AS/R systems can also be used to store loads double-deep in the racks. To this end, the cranes can be equipped with double-deep telescopic forks. Deep lane, or compact, multi-deep (3D) AS/R systems can store loads even more deep in storage lanes (see Figure 2.5). The storage depth depends on the type of product and the technology; e.g., 5-15 loads. These systems are particularly popular for storing products when storage space minimization is a primary concern, e.g., fresh produce and cold storage warehouses. In a typical crane-based compact storage system, a storage and retrieval (S/R) crane takes care of movements in the horizontal and vertical directions of the rack, and an orthogonal conveying mechanism takes care of the depth movement. Multi-deep lane crane-based compact storage systems can be further classified into three categories based on the mechanism of the depth movement: push-back rack, conveyor-based, and satellite-based (see Figure 2.2).

Figure 2.5: A crane-based multi-deep compact storage system (De Koster et al., 2008)

Push-Back Rack: In this variant, the crane (or automated forklift) stores the loads by mechanically pushing them into the storage lanes. The system works according to the Last-In-First-Out (LIFO) principle. A slight slope on the storage lane utilizes the gravity to ensure that a load is always available in front of the storage lane. The depth of the lane in a push-back pallet rack is up to about five loads.

Conveyor-Based: The racks in these systems are equipped with conveyors (see Figure 2.6). If the conveyor can move in two directions, the operation is LIFO, similar to the push-back racks. The conveyors can also operate in pairs (either by gravity or powered). On the inbound conveyor, unit loads flow to the rear end of the rack. The outbound conveyor is located next to the inbound conveyor. On the outbound conveyor, unit loads flow to the rack's front end and stop at the retrieval position of the crane. In the case of a gravity conveyor, the rack is equipped with a simple elevating mechanism at the back of the rack to lift unit loads from the down inbound conveyor to the upper outbound conveyor (see Figure 2.6). A stop switch located at the front side of the outbound conveyor stops a unit load when it is needed for retrieval. The lift drives the rotation of unit loads and, as it is the slowest element, it determines the effective rotation speed. In order to retrieve a pallet, the two neighboring gravity conveyors should have at least one empty slot (De Koster et al., 2008). The system with powered conveyors does not need lifts, but uses more expensive powered conveyors (that are not so easy to fix in the case of a malfunction). However, powered conveyors allow more dense storage because racks with powered conveyors can be constructed deeper than racks with gravity conveyors.

Satellite-Based: In this variant, a satellite (connected to the crane) or a shuttle (freely roaming) is used to perform the depth movement. The crane with a shuttle picks up a storage pallet and travels to the storage lane. Then the crane releases the shuttle in the rack and

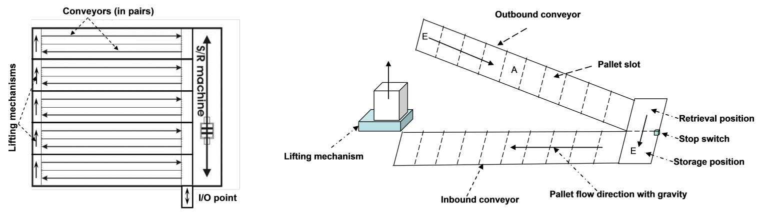


Figure 2.6: Working mechanism of gravity conveyor (De Koster et al., 2008)

the shuttle travels along the storage lane to store the load. Likewise, to retrieve a load, the shuttle travels underneath the load to retrieve the pallet and completes the remaining operations in a reverse sequence. In some cases, the shuttles can also be dedicated to lanes. If a system has fewer shuttles than storage lanes, the crane moves the shuttles between the lanes (Stadtler, 1996).

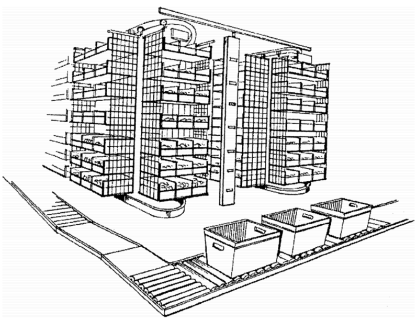
Unlike single-deep AS/R systems, the number of papers on multi-deep AS/R systems is limited. Sari et al. (2005) develop closed-form travel time expressions for a flow rack AS/RS. The expressions, which rely on a continuous storage rack approximation, are validated using discrete-even simulations. The simulations use a discrete rack dimensional approach. They find that the percentage errors are quite reasonable (varying between 11%-14%). Hence, such models can be used to estimate system throughput capacity.

De Koster et al. (2008) develop closed-form travel time expressions for a crane-based compact storage system with rotating conveyors, using a single-command cycle and random storage policy. The crane's expected retrieval travel time is identical for both gravity and powered conveyors. Using the expected travel time expressions, they calculate the optimal ratio between the three dimensions that minimized the travel time. They also provide an approximate travel time expression for dual command cycles and use it to optimize the system dimensions. They find a counter-intuitive result that the cube-in-time dimensions for the rack is not the optimal choice. The performance for a cube-in-time rack is still fairly good and deviates from the optimal rack configuration (optimal ratio along the three dimensions: 0.72:0.72:1) by about 3%. Yu & De Koster (2009b) extend the analysis of De Koster et al. (2008) for a turnover-based storage policy and determine the optimal rack dimensions that minimizes the expected cycle time. They analytically determine the optimal rack dimensions for any given rack capacity and ABC curve skewness. They find that with greater skewness of the ABC curve, savings in the expected time increase compared to the random storage policy. Yang et al. (2015) further extend the analysis of De Koster et al. (2008) by optimizing the shape of the system and by considering the acceleration and deceleration of the S/R machine, which has a direct impact on the optimal shape of the system. For the special case of constant speed of the S/R machine, their findings are in line with the results of De Koster et al. (2008). Hao et al. (2015) also develop expected travel time expressions and optimize the rack layout for a random storage policy. However, they choose an I/O point located in the middle of the rack (which, in reality, is difficult to construct for aisle-captive cranes). Under the same operating conditions, they obtain lower expected travel time and higher throughput.

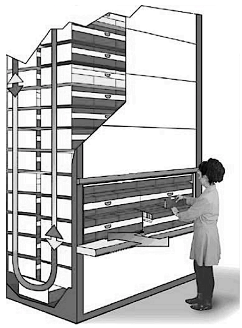
One of the biggest disadvantages of dense storage is that the pallets are accessible from only one side. Therefore, pallets are either retrieved based on LIFO principle or they undergo multiple relocations/reshuffles to allow access to the right pallet. Stadtler (1996) uses the retrieval time estimate of each pallet and proposes a storage and retrieval assignment planning tool considering this issue. The decision models are formulated as mixed-integer programs and are solved using a tabu search heuristic. The results show that the compact storage systems can operate at heavy workload and high storage rack utilization with a small number of pallet relocations (6% relocations at 78% rack utilization over a period of 42-day operation). Yu & De Koster (2012) develop heuristic approaches to sequence a block of storage and retrieval transactions for a compact conveyor-based storage system operating in a dual-command cycle. They compare the makespan performance for five sequencing heuristics: 1) First Come First Serve (FCFS), 2) Nearest Neighbor (NN), in which the sequence is based on the minimum travel distance between storage and retrieval locations, 3) Shortest Leg (SL), in which the open storage location lies on the Tchebychev path leading to the retrieval location, 4) Shortest Dual Cycle (SDC) in which sequencing is done in a way to minimize the dual cycle time in every step, and 5) Percentage Priority to Retrievals with Shortest Leg (PPR-SL), in which a certain percentage of retrievals are given a higher priority for pre-positioning than the storage open locations. Numerical results suggest that PPR-SL strategy outperforms all sequencing strategies by 20% or more. For a compact AS/R system with shuttles or satellites, one of the biggest challenges is the additional time required to reshuffle unit loads and retrieve the right unit. Many companies, therefore, use a dedicated storage policy per lane, which reduces the reshuffle time, but decreases lane utilization (and requires a larger system). To overcome this shortcoming, Zaerpour et al. (2013) propose a mathematical model for a shared storage policy that minimizes the total retrieval time in a cross-dock/temporary storage environment. They solve the model using a construction and improvement (C&I) heuristic. They show that for most real cases, shared storage outperforms dedicated storage, with a shorter response time and better lane utilization. Yu & De Koster (2009c) focus on identifying the optimal class zone boundaries for a compact 3D crane-based systems with two storage classes (a high turnover class and a low turnover class). They formulate the problem as a non-linear integer program and obtain a solution using a decomposition technique and a one-dimensional search scheme. They show that the crane travel time is significantly influenced by zone dimensions, zone boundaries, and the ABC curve skewness.

2.4 Carousels, Vertical Lift Modules and Automated Dispensing Systems

Carousels are automated storage and retrieval systems in which shelves are linked together and rotate in a closed loop. The rotation is either horizontal or vertical (see Figure 2.7a and Figure 2.7b). In this system, the picker has a fixed location in front of the system, and the system transports the items to the picker. Carousels are especially suitable for small and mid-size items such as books, health and beauty products (Litvak & Vlasiou, 2010).



(a) Horizontal carousel



(b) Vertical carousel

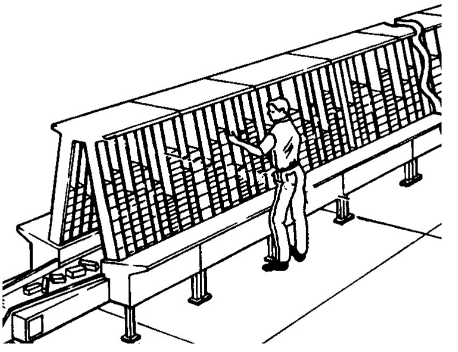
Figure 2.7: Carousels (Meller & Klote, 2004; Litvak & Vlasiou, 2010)

A Vertical Lift Module (VLM) is similar to a carousel, but operates differently. It consists of two columns of trays with a lift-mounted inserter/extractor in the center (see Figure 2.8a). When an item is needed, the inserter/extractor locates the trays in which the item is stored and brings the tray to the picker, who is located in front of the system, like in a carousel (MHI, 2015). The static location of the picker in these systems eliminates pickers walking (Meller & Klote, 2004), which can improve picking productivity. The pickers can also perform other tasks such as packing and labeling or even serving another carousel or VLM while waiting for the carousel to retrieve items.

In an automated dispensing system, products are dispensed automatically. The replenishment is still carried out manually, but it can be done without interrupting the picking process. A common automated dispensing system is the A-frame. This system consists of product channels positioned in an “A” shape layout creating a tunnel in which the collection belt is located. The orders are filled by automatically dispensing the corresponding products in a virtual window on the conveyor belt (see Figure 2.8b). A-frames are suitable for large orders of small-sized items. The systems are mainly used in pharmaceutical, cosmetic and mail order industries (Pazour & Meller, 2011; MHI, 2015).

Horizontal carousel models have been extensively studied in the literature dating back to the 1980s when the basic foundation for studying carousals was laid out by Bartholdi III &

(a) Vertical lift module



(b) A-Frame

Figure 2.8: Vertical lift module and A-frame (MHI, 2015, 2017)

Platzman (1986). Different aspects have been studied such as storage arrangement, response time, and design issues. Litvak & Vlasiou (2010) give an extensive literature overview on performance evaluation and design of carousel systems. Pazour & Meller (2013) investigate the effect of batch retrieval on the performance of the horizontal carousel system. They show that batching retrievals reduces the cycle time in the carousel by 20% compared to sequential processing. The number of studies on horizontal carousels have declined and the only recent study is by Pazour & Meller (2013). The reason could be that more and more companies are replacing their horizontal carousels with shuttle-based storage and retrieval systems, which we discuss in Section 2.5. VLMs, on the other hand, have been studied only in a handful of articles. Meller & Klote (2004) develop a throughput model for a single VLM pod. Dukic et al. (2015) extend the research to model the throughput of a dual-tray VLM. Rosi et al. (2016) use simulation to analyze the throughput performance of the single-tray VLM for different design profiles (height and width of VLM) and the lift velocity. Similar to VLMs, A-frames and automatic dispensers have been studied in few articles. Caputo & Pelagagge (2006) develop a decision support system for an A-frame system. They use a heuristic approach to determine the number of channels in the system, reorder level and maximum quantity to be dispensed based on recorded performance of the last period and the forecasted demand. Meller & Pazour (2008) investigate an SKU assignment problem for an A-frame, and use a knapsack heuristic approach to solve it. Pazour & Meller (2011) develop a mixed-integer linear program to determine the infrastructure investment of an A-frame as well as SKU allocation to the A-frame. They develop a heuristic solution to a solve real size SKU allocation problem. They also propose a closed-form equation to calculate the system throughput of an A-frame. Imahori & Hase (2016) investigate the SKU assignment of an A-frame as well as the optimal sequencing of the order retrievals in order to minimize the total retrieval time. They analyze the problem for computational complexity and develop a graph-based heuristic to obtain the best sequence for retrieving orders and SKU allocations. Kim et al. (2016) study the effect of different ejecting zone (EZ) methods on the performance of an A-frame system. An EZ is a segment of the conveyor belt that is dedicated to an order on which the required SKUs for that specific order are ejected while that zone passes through the A-frame. They investigate three EZ methods: unequal, equal, and combined. They use simulation to show which EZ method is suitable for the system, depending on the order throughput time and energy usage.

2.5 Aisle-based Shuttle Systems

Throughput capacity of AS/R systems is constrained because only one crane is responsible for handling loads at all vertical levels within a given storage aisle. This led to a new generation of automated order picking systems, Autonomous Vehicle-based Storage and Retrieval Systems (AVS/RS), which were first introduced by Savoye Logistics in the 1990s. Such systems are increasingly popular because the required investment is similar to that of AS/R systems, while they offer a much higher retrieval capacity, and are also significantly more flexible in capacity. By using additional shuttles, system capacity can be increased, and by removing shuttles, capacity can be decreased. Typical AVS/R systems use shuttles, which can drive in the x-direction and the y-direction on any level in the aisle, and lifts move shuttles (or unit loads) between the levels. In this variant, shuttles can only move horizontally, and rely on lifts for vertical movements. Recently, several robotic solutions have emerged, in which the shuttles (called robots) have the ability to not only move horizontally but also to elevate up to different tiers by either moving diagonally or vertically (Azadeh et al., 2019b). Therefore, the AVS/R system can be classified based on their shuttles' movement capability into three categories: *Horizontal*, *Vertical*, and *Diagonal* systems (see Figure 2.2). In this section, we discuss different types of Horizontal systems and leave the discussion on Vertical and Diagonal systems for Section 2.8.

2.5.1 System Description

Single/Double-Deep Storage

The storage area in an AVS/R system consists of aisles with multi-tier storage racks on both sides and a cross-aisle that runs orthogonal to the aisles. To perform storage and retrieval actions, a lift is used for vertical movements between tiers and autonomous vehicles or shuttles are used for the horizontal movements within the tier (Roy, 2011). To retrieve a tote, a shuttle moves to the tote's storage location and picks up the tote, pulls it on board and moves towards the lift for vertical travel. Then the shuttle either hands the tote to the lift (tier-captive system (Heragu et al., 2008)), or uses the lift to move the load to a lower level (tier-to-tier system (Heragu et al., 2008)) where it is transferred to the pick station by conveyor belt. After picking, the tote again uses the lift and a shuttle to be stored in the system.

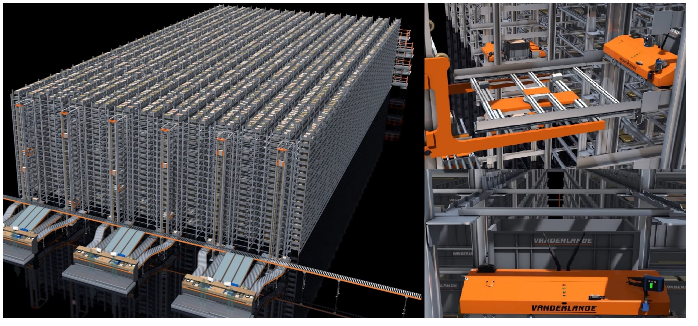


Figure 2.9: Adapto$^{\mathrm{TM}}$AVS/R system (Source: Vanderlande)

Multi-Deep (Compact) Storage

Crane-based compact storage systems lack flexibility in the volumes they can handle. Shuttle-based multi-deep storage systems, using lifts instead of cranes, have more throughput flexibility by adding or removing shuttles. They are adapted for safe and secure handling of a variety of products such as textiles, automobile spare parts, and fresh produce.

These systems consist of multiple tiers of multi-deep storage lanes, each of which holds one type of product (see Figure 2.10). The loads in a lane are managed using a last-in-first-out (LIFO) policy unless the retrieval is possible from opposite sides. In such a system, the vertical transfer of loads (usually pallets) across multiple tiers is carried out using lifts, whereas the horizontal transfer of loads within a tier is carried out using shuttles. These shuttles move underneath the loads within each storage lane to store or retrieval the load.

The horizontal movements of shuttles and loads in the system can be carried out either by "specialized" shuttles and a transfer car, or by "generic" shuttles that can move in both horizontal directions without the transfer car.

2.5.2 Literature

Single/Double-Deep Storage

Using the framework we discuss in Section 2.2.2, the literature on the single/double-deep horizontal AVS/R systems is categorized in three categories: *System Analysis, Design Optimization, and Operations Planning and Control*.

Figure 2.10: Multi-deep shuttle-based compact storage system (Tappia et al., 2016)

System Analysis: Malmborg (2002) was the first to analyze the AVS/R system. He developed a state equation model to estimate the vehicle utilization and cycle time of the unit-load AVS/RS. He estimates the vehicle cycle time to be $(1-\alpha)t_{SC} + \alpha t_{DC}/2$, in which t_{SC} and t_{DC} denote the single-command and dual-command cycle times and α is the proportion of all cycles that are dual-command cycles. Malmborg (2003a) emphasizes the design advantage of an AVS/R system relative to an AS/R system, which is the ability to adapt the vehicle fleet size in response to the transaction demand. Malmborg (2003b) extends the state equation model by including the number of pending transactions in the state space description, to estimate α, in a system with opportunistic interleaving, i.e., dual-command cycles are used only if storage and retrieval requests are pending in the transaction queue at the time when the cycle is initiated. However, the state equation approach is computationally inefficient for solving large scale problems. Therefore, Kuo et al. (2007) and Fukunari & Malmborg (2008) propose a computationally efficient model to overcome this problem. In this approach, the lift system is modeled as a closed queuing network which is nested within a separate vehicle closed queuing network. They model the queuing dynamics between vehicles and transactions using an $M/G/V$ queue (with V vehicles), and the dynamics between transactions/vehicles and lift using a $G/G/L$ queue (with L lifts). The two systems are analyzed iteratively until the performance measures converge. Although the nested queuing approach is computationally efficient, it is not able to model a scenario in which the cycle starts outside of the storage rack, i.e., when loads are received from outside the storage rack. Fukunari & Malmborg (2009) propose a queuing network model as an alternative to address this drawback. They propose a closed queuing network for estimating resource utilization in the AVS/R systems. Although the earlier models are effective in estimating vehicle utilization with reasonable accuracy, they are ineffective in estimating transaction waiting times. Using a series of queuing approximations, Zhang et al. (2009) address this problem by dynamically choosing among three different queuing approximations, based on the variability of transaction inter-arrival times. This procedure significantly improves the accuracy of transaction waiting time estimates. Recent studies use a semi-open queuing network to analyze the performance of the AVS/R system and estimate the external transaction waiting time with better accuracy. Roy et al. (2012) build a multi-class SOQN with class switching for a single-tier AVS/RS, and design a decomposition method to estimate system performance. Ekren et al. (2013) model a tier-to-tier AVS/RS as an SOQN and present an analytical approximation by extending the algorithm of Ekren & Heragu (2010a) to estimate the performance measures. Later, Ekren et al. (2014) improved the estimation of the number of transactions waiting in the vehicle queue by developing a matrix-geometric method for the SOQN model. Cai et al. (2014) model a tier-to-tier system as a multi-class multi-stage SOQN, and use matrix-geometric methods to analyze it. Ekren (2011) performs a case study by simulating the performance of a real AVS/RS under pre-defined design scenarios (number of aisles, bays, tiers, and vehicles). He also includes the total cost of the system in his analysis. The number of studies on tier-captive configurations is limited. Heragu et al. (2011), Marchet et al. (2012), and Epp et al. (2017) use the open queuing network approach to estimate the transaction cycle time of the AVS/R system with tier-captive vehicles. Heragu et al. (2011) then use an existing tool called the Manufacturing Performance Analyzer (MPA) to compare the performance of AVS/R systems and traditional AS/R systems. Ekren (2017) uses simulation to model the system and provide a graph-based solution for performance evaluation of the system (utilization of lifts and the cycle time) under various design configurations. Roy et al. (2017) model the system as an integrated queuing network and estimate the cycle time and resource utilization. They model each tier as a semi-open queuing network and the vertical transfer unit as a multi-class queuing network with $G/G/1$ queues corresponding to each vertical transfer segment. They replace each tier subsystem with a single load-dependent queue, and approximate the first and second moments of inter-departure times using embedded Markov chain analysis. Then they solve the integrated model by capturing the linkage between arrivals and departures in the tier subsystem and the vertical transfer unit. Lerher et al. (2015) and Lerher (2016) develop travel time models for single-deep and double-deep AVS/R systems, respectively. They develop a closed-form expression for the cycle time and consider the effect of shuttle acceleration and deceleration.

Design Optimization: Roy et al. (2012) develop a semi-open queuing network model and optimize the shape of the system. Their results suggest that the layout configuration with depth-to-width ratio $D/W = 2$ for a system with the lift in the middle, provides the best system performance. Roy et al. (2015a) extend the model, and show that the end of the aisle is the optimal cross-aisle location for the system. Ekren & Heragu (2010b) provide a simulation-based regression analysis for the rack configuration of the system. In their regression model, the average cycle time is chosen as the output variable, and the input variables are the number of tiers (T), aisles(A) and bays (B). The regression function demonstrates that the cycle time is positively related to T and B, but is negatively related to $T * A$ as well as to $T * B$. Marchet et al. (2013) simulate an AVS/R systems with a tier-captive configuration and illustrate the effect of rack configurations on the throughput performance. By varying the rack configuration and observing the performance impact, they optimize the shape of the system.

Operations Planning and Control: Ekren et al. (2010) develop a simulation-based experimental design to identify the effect of a combination of several input factors (dwell-point policy, scheduling rule, I/O location, and interleaving rule) on the performance of the system (average cycle time, average vehicle, and lift utilization). They investigate the effect of up to four-way interactions of input variables on the performance of the system. Kuo et al. (2008) use the closed queuing network approach to investigate the effect of a class-based storage policy on the cycle time of an AVS/R system. They conclude that class-based storage policies can mitigate the cycle time inflation effect of vertical storage, while keeping the space efficiency of the random storage intact. Kumar et al. (2014) simulate an AVS/R system in which the vehicles are captive in vertical zones rather than in tiers. They show that the optimal partitioning of vertical zones can reduce the transaction cycle times by up to 12% compared to the tier-captive configuration. Roy et al. (2012) develop a semi-open queuing network model and analyze the effect of vehicle location, the number of storage zones, and vehicle assignment policies on the performance measures. They show that using multiple zones reduces travel time along the cross-aisle which improves the performance of the system. However, increasing the number of zones beyond a threshold results in longer transaction waiting time and worsens the system performance. Finally, they observe that the most efficient vehicle assignment policy is the random policy. Roy et al. (2015a) extend the model to analyze different dwell-point policies. They shows that the best dwell policy is the L/U point dwell policy. He & Luo (2009) use colored time Petri nets to dynamically model AVS/R systems and established the necessary conditions to have a deadlock-free system. Roy et al. (2014) use a semi-open queuing network to investigate the effect of vehicle blocking within a single tier of the AVS/R system. Their results show that the blocking delays could contribute significantly (up to 20%) to the transaction cycle time. They also show that the percentage of blocking delays goes up as the number of vehicles increases. However, the effect of blocking decreases as the utilization of vehicles increases, since the waiting time to obtain a free vehicle dominates in a system with high vehicle utilization. Roy et al. (2016) arrive at a similar conclusion using a simulation model. Roy et al. (2015b) evaluate congestion effects in a multi-tier AVS/R system. They develop a semi-open queuing network and use a decomposition-based approach to solve it. Their model provides the steady state distribution of the vehicles at the cross-aisles and aisles of each tier, conveyor loops, at the LU point. The model also captures the blocking delays at the cross-aisle and aisle nodes. Zou et al. (2016) investigate a scenario in which the lift and vehicles in the tier-captive AVS/R system are requested to move a load simultaneously rather than sequentially. They model the system with a fork-join queuing network. They show that the parallel processing policy improves the response time of the system by at least 5.5% compared to the sequential processing policy, for small-sized systems (system with fewer than ten tiers). In large systems with more than ten tiers and a ratio of aisle length to rack height of more than seven, they find a critical point for the retrieval transaction arrival rate. Before that rate, the parallel processing policy performs better. For arrival rates more than the critical point, the sequential processing policy should be used.

Multi-deep (Compact) Storage

The number of research articles on multi-deep AVS/R systems is limited. The literature is categorized in two categories: *System Analysis and Design Optimization.*

System Analysis: Manzini et al. (2016) develop an analytical model to determine the travel time and travel distance for single and dual-command cycles for a layout configuration. D'Antonio et al. (2018) present an analytical model to calculate the cycle time and its standard deviation for a system.

Design Optimization: Tappia et al. (2016) model each tier and the vertical transfer mechanism using a multi-class semi-open queuing network and an open queue, respectively. They suggest that generic shuttles may reduce the total travel distance for storage and retrieval operations since additional shuttle movements in the cross-aisle without a load are not required. However, they argue that a specialized shuttle might be attractive from an economic perspective, since a generic shuttle is about twice as expensive as a specialized one. They also show that a single-tier system with a depth/width ratio of around 1.25 minimizes the expected throughput time. Manzini et al. (2016) calculate the optimal location of the L/U point and the optimal shape of the system. They also calculate the optimal number and depth of the lanes depending on the demand pattern by minimizing the operative costs and maximizing the storage space efficiency.

Table 2.2 presents an overview of the literature on shuttle-based storage and retrieval systems with aisles.

2.6 Grid-Based Shuttle Systems

In this section, we discuss a variant of the shuttle-based automated storage and retrieval systems in which shuttles move on a grid. In a grid-based system, the storage locations are either dynamic or static (see Figure 2.2). In a dynamic storage system (or puzzle-based system), the stored SKUs need to move (on a shuttle) in order to store or retrieve an item. We discuss the static storage systems in Section 2.8.

Table 2.2: Overview of the literature (34 papers) on shuttle-based storage and retrieval system (aisle-based)

Research Category	System	Article	Research Issue	Methodology
System Analysis	Single/Double Deep (tier-to-tier)	Malmborg (2002), Malmborg (2003a,b)	Estimate vehicle utilization and cycle time	State equation model
		Kuo et al. (2007), Fukunari & Malmborg (2008)	Estimate vehicle utilization and cycle time	Nested queuing model
		Fukunari & Malmborg (2009)	Estimate vehicle utilization and cycle time interfacing material flow system	Closed queuing network
		Zhang et al. (2009)	Estimate transaction waiting time	Variance-based nested queuing model
		Ekren (2011)	Evaluate performance of a real system under predefined design scenarios	Simulation
		Ekren et al. (2013), Ekren et al. (2014), Cai et al. (2014)	Model the system	Semi-open queuing network
	Single/Double Deep (tier-captive)	Marchet et al. (2012), Epp et al. (2017)	Estimate transaction cycle time	Open queuing network
		Heragu et al. (2011)	Estimate transaction cycle time, Compare with AS/RS	Open queuing network network
		Lerher et al. (2015), Lerher (2016)	Estimate mean travel time	Closed-form solution
		Ekren (2017)	Graph-based solution for performance evaluation of the system	Simulation
		Roy et al. (2017)	Estimate transaction cycle time and resource utilization	Multi-stage semi-open queuing network
	Multi-Deep	Manzini et al. (2016), D'Antonio et al. (2018)	Estimate cycle time	Travel time model
Design Optimization	Single/Double Deep (single tier)	Roy et al. (2012)	Optimal rack configuration	Semi-open queuing network
		Roy et al. (2015a)	Optimal cross-aisle location	Semi-open queuing network
	Single/Double Deep (tier-to-tier)	Ekren & Heragu (2010b)	Optimal rack configuration	Simulation-based regression
	Single/Double Deep (tier-captive)	Marchet et al. (2013)	Optimal rack configuration of the system	Simulation
	Multi-Deep	Manzini et al. (2016)	Optimal L/U point location, Optimal layout configuration, Optimal number and depth of the lanes	Semi-open queuing network
		Tappia et al. (2016)	Optimal layout configuration, choice of shuttle and vertical transfer	Semi-open queuing network
Operations Planning and Control	Single/Double Deep (single tier)	Roy et al. (2012)	Effect of design choices on cycle time and vehicle utilization	Semi-open queuing network
		Roy et al. (2014)	Effect of vehicle blocking on performance	Semi-open queuing network
		Roy et al. (2015a)	Optimal dwell-point policy	Semi-open queuing network
		Roy et al. (2016)	Effect of vehicle blocking on performance	Simulation model
	Single/Double Deep (tier-to-tier)	Kuo et al. (2008)	Effect of class-based storage on cycle time	Closed queuing network
		He & Luo (2009)	Deadlock-free control policy	Colored time Petri nets
		Ekren et al. (2010)	Effect of combination of dwell-point, I/O location, scheduling and interleaving rule on performance	Simulation, ANOVA
		Kumar et al. (2014)	Optimal partitioning of vertical zones in the system	Simulation
		Roy et al. (2015b)	Congestion effect on the performance of the system	Semi-open queuing network
	Single/Double Deep (tier-captive)	Zou et al. (2016)	Simultaneously vs sequentially requesting vehicles and lifts	Fork-join queuing network

2.6.1 System Description

Gue (2006) shows that the storage density of a k-deep aisle-based system, is less than or equal to $2k/(2k + 1)$, i.e., 2/3 for a single-deep and 4/5 for a double-deep system. To achieve an absolute maximum storage density, a new concept based on the famous Sam Loyd's puzzle game has been developed; the 15-slide puzzle (Loyd & Gardner (1959)). The 15-slide puzzle is a game in which 15 numbered tiles slide within a 4×4 grid, and the objective of the game is to arrange the tiles in the correct numerical sequence, starting from a random initial arrangement.

The "Puzzle-Based Storage and Retrieval" concept (Gue & Kim, 2007), follows a similar idea. A tile represents a tote, a pallet, or even a container that is stored in a grid with only one open spot on the grid, which allows a $(n - 1)/n$ storage density, where n is the number of cells in the grid. To retrieve a requested unit load, the system repeatedly moves the open locations, which ultimately brings the load to the Input/Output (I/O) point. This is illustrated in Figure 2.11.

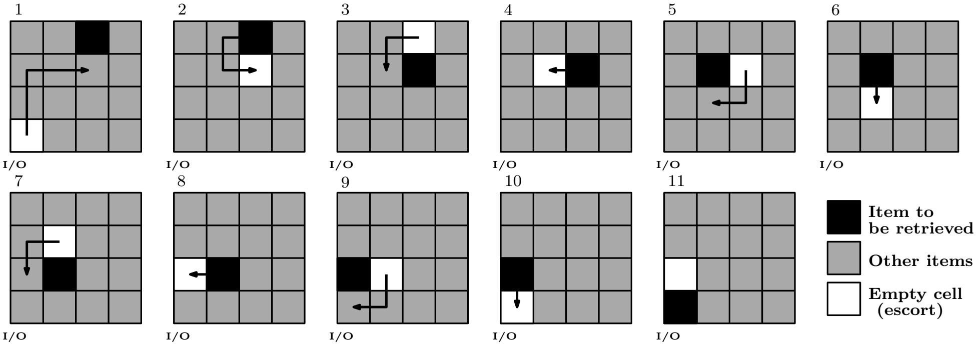


Figure 2.11: Maneuvering a load (item) to the I/O point

To retrieve a load, an open location first needs to be moved next to the requested item. Then the open location should be used to move the item to the I/O point. In other words, the open location "escorts" the requested item to the I/O point. An open location is called an *escort* (Gue & Kim, 2007). Several compact storage system variants have emerged from the puzzle-based concept in practice and in the literature.

GridStore: Building upon the puzzle-based storage system concept, Gue et al. (2014) propose a high-density storage system for physical goods called GridStore. The system consists of a rectangular grid of square conveyor modules with the capability to move items in the four cardinal directions. The modules can communicate with their neighboring modules as well as with the item they carry. At the south side of the grid, the retrieval conveyor moves products away from the grid. At the north a replenishment conveyor moves products that need to be stored in the grid. Figure 2.12 illustrates the movement of the items toward the retrieval conveyor.

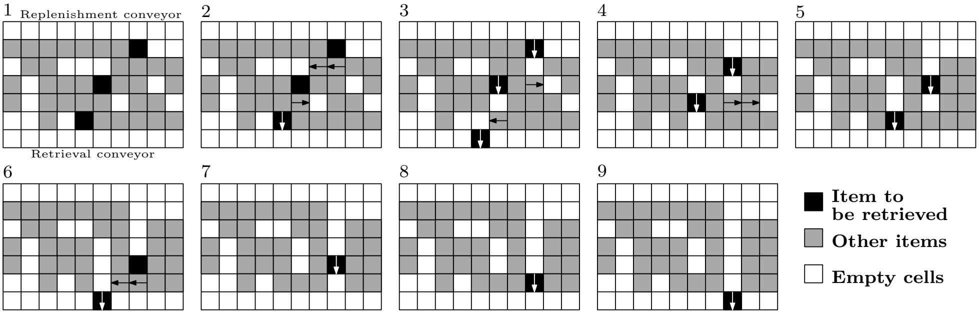


Figure 2.12: Items movements toward the retrieval conveyor in GridStore

GridPick: Based on the GridStore architecture, Uludag (2014) introduces an order picking systems called GridPick. The system is filled with high-density storage containers, without any fixed lanes or aisles; only a few open spots on the grid allow items to move during the retrieval process. The objective of the system is to provide a high order picking rate while minimizing any congestion effects. Unlike the GridStore, items do not leave the grid in the GridPick system. Only containers holding the requested item, move to the edge of the system, called the pick face. The picker picks the items and accumulates the order in a picking cart. There is also a backward movement, away from the pick face, to balance the empty cells in each row. This balancing rule helps to avoid deadlocks in the system.

Figure 2.13 illustrates an instance of GridPick. The gray items are not-requested stored items, and the black items are the requested items which are moving toward the pick face. Black circles on top of some gray items are balancing items moving in the opposite direction from the pick station. The numbers on top of the items display the order number for the requested item. The next order for picking is released when all the items from an order have arrived in the pick face of the system.

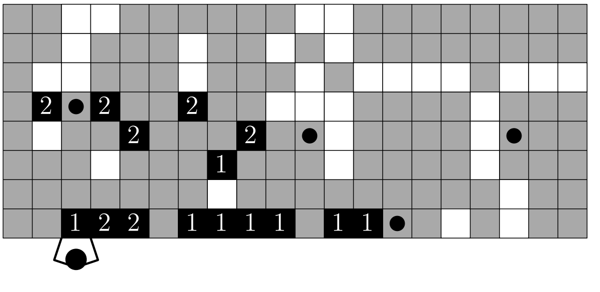


Figure 2.13: An instance of the GridPick system

When comparing the GridPick with its equivalent gravity flow rack counterpart, Gue & Uludag (2012) show that the gravity flow rack results in a larger system. Therefore, the average productivity (measured in picks per hour) is higher for smaller orders in the GridPick because it reduces travel. However, as order size increases, walking time of both systems converges to the same number.

Live-Cube Compact Storage: A multi-level system in which each floor is based on a puzzle-based storage architecture is called a Live-Cube storage system (Zaerpour et al., 2015). As illustrated in Figure 2.14, the essential parts of the system are multiple levels of storage grids, shuttles, lifts, and the I/O points. Each level of the system forms a grid-based storage system where shuttles move in x and y directions with the load on top of them. With at least one escort available in each level, the shuttles maneuver the requested item to the lift, which transports the load to the I/O point. The I/O point is usually located at the lower left corner of the system.

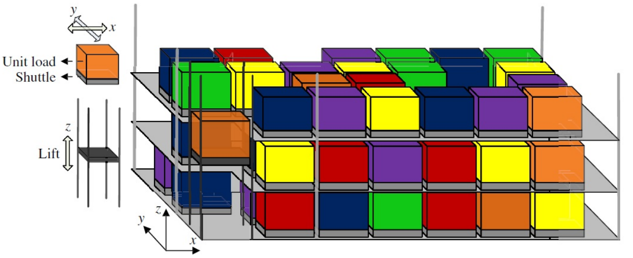
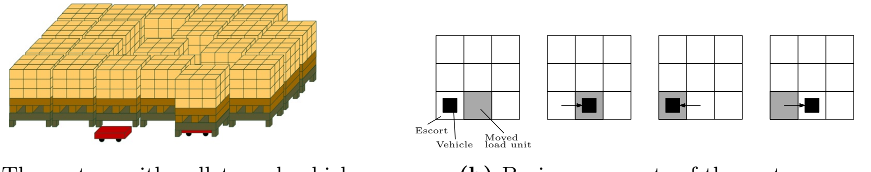


Figure 2.14: A Live-Cube storage system with lift (Zaerpour et al., 2015)

GridFlow System: A major drawback of the puzzle-based system is that the physical layout cannot be changed easily. Therefore, the concept of GridFlow is proposed by Furmans et al. (2011) to offer a cheaper and a more flexible system. In this system, instead of conveyors, Automated Guided Vehicles (AGVs) are used to move the pallets. The use of AGVs instead of conveyors makes the system more flexible with respect to design and throughput changes. Vehicles can form grids of any shape without any additional investment. Figure 2.15 illustrates the GridFlow system and the vehicle movements in the system.



(a) The system with pallets and vehicles **(b)** Basic movements of the system

Figure 2.15: The GridFlow system (Furmans et al., 2011)

Many system manufacturers are developing puzzle-based systems in different variants. The number of actual implementations and prototypes based on this concept is growing in many different fields, especially in the automated parking systems (e.g., “Park, Swipe, Leave” parking system (Automation Parking System, 2016), “Space Parking Optimization Technology” or SPOT (EWECO, 2016), Hyundai Integrated Parking System or HIP (Hyundai Elevator Co. LTD., 2016), Wöhr Parksafe (Wöhr, 2016)).

2.6.2 Literature

The literature on the puzzle-based storage and retrieval systems is categorized in three categories: *System Analysis, Design Optimization, and Operations Planning and Control.*

System Analysis: Gue & Kim (2007) develop an algorithm to find an optimal path to retrieve an item in the puzzle-based system with a single escort positioned at the I/O point. They propose a dynamic programming approach for multiple escorts and a heuristic for larger instances. Their results confirm the intuition that having more escorts shortens the retrieval time. The only exception occurs for smaller systems with many escorts at the I/O point. They also compare the performance of the puzzle-based system with its aisle-based counterpart. They find that aisle-based systems have shorter retrieval times than puzzle-based systems, unless the desired storage density is more than 90%. Kota et al. (2015) develop a closed-form expression for the retrieval time in the puzzle-based storage system with a single or two randomly scattered escorts within the grid. They propose a heuristic solution for more than two escorts in the system. Their heuristic gives a near optimal solution, except for the time when free escorts are congested near the edge of the grid. Zaerpour et al. (2015) investigate a multi-tier puzzle-based (live-cube) storage system. They assume that there are sufficient escorts available at each level so that a virtual aisle can be created (minimum number of escorts is the maximum of the rows and columns in the system). They use traditional methods for the aisle-based system and derive a closed-form formula for expected retrieval time. Zaerpour et al. (2017b) propose a two-class-based storage policy for a live-cube system. They derive closed-form formulas to calculate the expected retrieval time of the system. They conclude that their proposed storage policy can improve the average response time of the system up to 55% compared to the random storage policy, and up to 22% compared to the cuboid two-class-based storage policy.

Design Optimization: Gue et al. (2014) analyze the optimal shape of the GridStore system. They find that a system with more columns has a higher throughput with the same number of stored items. Zaerpour et al. (2015) propose and solve a mixed-integer-nonlinear model to optimize the dimensions of a live-cube system by minimizing the retrieval time assuming a random storage policy. Zaerpour et al. (2017b) extend this work by considering a two class-based storage policy. Their results show that the the optimal dimensions of the system are identical for two class-based and for a random storage policy. Zaerpour et al. (2017a) propose a mixed-integer nonlinear model to optimize the dimensions and zone boundaries of the two-class live-cube storage system by minimizing the response time. Furmans et al. (2011) investigate the design choices for the GridFlow system with one vehicle and one escort. They conclude that putting the I/O point in the middle of the longer side of the grid produces the best performance. Furthermore, they show that the 2:1 aspect ratio results in the lowest retrieval time when the number of storage locations is less than 2000. Their results are not conclusive for larger storage capacities.

Operations Planning and Control: Taylor & Gue (2008) investigate the effect of the distribution of escorts in the puzzle-based system. They examine three choices for the initial location of escorts: 1) near the I/O (located at a lower left corner of the grid), 2) along the diagonal from lower left to upper right, and 3) randomly on the grid. They show that when the number of escorts is above 25%, having the escorts along the diagonal always outperforms the other strategies. The only exception occurs when the storage is based on an ABC policy, in which, random placement for the escorts is the best option. Yu et al. (2017) consider a puzzle-based storage system with multiple escorts, in which multiple loads and escorts are allowed to move simultaneously and in blocks (simultaneous movement of loads in a line). Using integer-programming, they obtain the optimal retrieval time of a single item in the system. Their results show that allowing loads and escorts to move simultaneously and in blocks can save up to 70% in the total number of needed moves to retrieve an item. Mirzaei et al. (2017) propose an approach for simultaneous retrieval of multiple items. They derive the optimal retrieval time for double-item and triple-item retrieval using enumeration. They propose a heuristic algorithm for more than three simultaneous item retrievals. They show that double-item retrieval policy reduces the storage/retrieval time by an average of 17% compared to a sequential retrieval policy. Cycle time savings can be further increased by performing multi-item retrievals. Gue et al. (2014) propose a decentralized Assess-Negotiate-Convey control scheme for the GridStore system, in which each conveying cell can execute the same set of instructions based on its local condition. They also investigate the effect of WIP and the number and distributions of escorts per row, on the throughput. First, they assume that escorts are uniformly distributed in the rows. They show that for medium and low level of WIP in the system, the throughput increases with an increasing rate with an additional request. Next, they investigate two additional distribution of escorts: more escorts in the southern row (increasing k) and fewer escorts in the southern row (decreasing k). They show that the distribution of escorts has no effect on the throughput for low level of WIP. The increasing k performs better at low to moderate WIP levels, and all distributions perform equally well at a high level of WIP. Alfieri et al. (2012) investigate the GridFlow system with a limited number of vehicles. They propose a heuristic algorithm to optimize the movement of shelves and to dispatch the AGVs optimally.

Table 2.3 presents an overview of the literature on the puzzle-based storage and retrieval systems.

Table 2.3: Overview of the literature (11 papers) on puzzle-based storage and retrieval systems

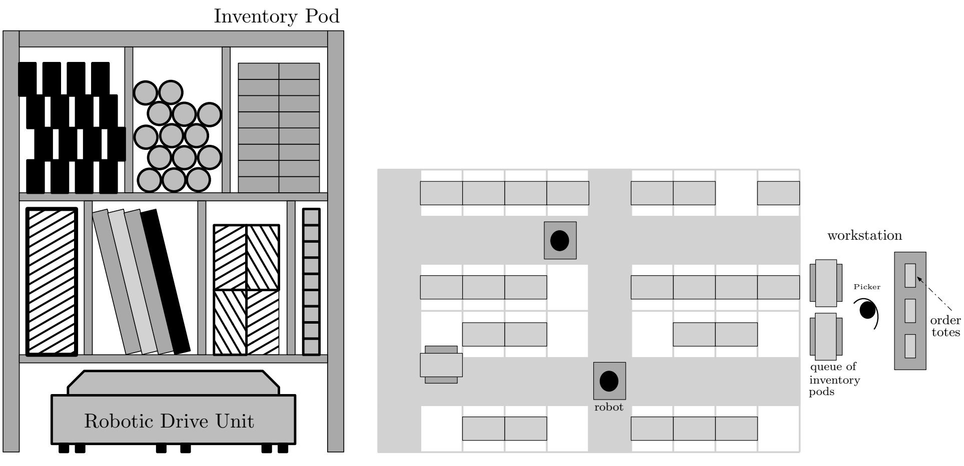
Research Category	System	Article	Research Issue	Methodology
System Analysis	Puzzle-Based	Gue & Kim (2007)	Optimal retrieval path with fixed escort positions, performance comparison with aisle-based	Dynamic programming, heuristics
		Kota et al. (2015)	Retrieval time estimation with randomly located escorts	Closed-form expression, heuristics
	Live-Cube	Zaerpour et al. (2015)	Retrieval time expression with random storage policy	Closed-form expression
		Zaerpour et al. (2017b)	Retrieval time expression with two class-based storage policy	Closed-form expression
Design Optimization	Puzzle-Based	Taylor & Gue (2008)	Effect of escort locations	Discrete time simulation
	Live-Cube	Zaerpour et al. (2015)	Optimal shape of the system with random storage policy	Mixed-integer nonlinear model
		Zaerpour et al. (2017b)	Optimal shape of the system with two-class-based storage policy	Close-form expression
		Zaerpour et al. (2017a)	Optimal zone boundary in two class-based storage policy	Mixed-integer nonlinear model
	GridFlow	Furmans et al. (2011)	Optimal shape of the system, choice of I/O point	Discrete time simulation
	GridStore	Gue et al. (2014)	Optimal shape of the system, effect of WIP and escorts on the throughput rate	Discrete time simulation
Operations Planning and Control	Puzzle-Based	Yu et al. (2017)	Effect of simultaneous and block movement of items and escorts	Integer-programming
		Mirzaei et al. (2017)	Simultaneous multi-load retrieval	Monte Carlo simulation, Heuristics
	GridStore	Gue et al. (2014)	Deadlock free decentralized control scheme, effect of WIP and escorts on the throughput rate	Discrete time simulation
	GridFlow	Alfieri et al. (2012)	Gridflow with limited number of vehicles, Optimally dispatch AGVs, Optimize the shelves' movement	Heuristics

2.7 Robotic Mobile Fulfillment Systems

Internet retailers typically have a warehouse with a large assortment of small products. Their demands usually consists of multi-line small quantity orders. In manual picking systems, much non-value added time is needed by the pickers to travel along the aisles. The Robotic Mobile Fulfillment System (RMFS) is a system, in which robots capable of lifting and carrying movable shelves retrieve the storage pods (i.e., movable shelf racks) and transport them to the pickers, who work in ergonomically designed workstations. Bringing the inventory to the picker instead of the picker traveling to the inventory, can double the picker productivity (Wurman et al., 2008). The system is also very flexible in throughput capacity, as more robots and pods can be added to the warehouse. This is particularly important for internet retailers who face volatile demand. The RMFS was conceptualized by Jünemann (1989) and was U.S. patented by KIVA Systems Inc. (Mountz et al., 2008), which then was acquired by Amazon and rebranded to AmazonRobotics. Today the system is operational in many Amazon facilities. Meanwhile, other providers have also entered the market with mobile racks in combination with robots, such as CarryPick^{TM} by Swisslog, Butler^{TM} by GreyOrange, Scallog System^{TM}, and Racrew^{TM} by Hitachi (Banker, 2016).

2.7.1 System Description

The RMFS consists of three major components: *1) Robotic Drive Units:* These robots are instructed by the central computer to transport inventory pods to the workstation for restocking or for picking. Nowadays also decentrally (or locally) controlled systems exist. *2) Inventory Pods:* Pods are movable shelf racks that contain the stored products. Pods come in two standard sizes. Smaller pods are used for weights up to 450 kg and large pods are used for weights up to 1300 kg. *3) Workstation:* Ergonomically designed areas where human workers perform pod replenishment, picking and packing functions (MWPVL International, 2012). Figure 2.16 presents an RFMS workstation and its components.



(a) Robot and inventory pod **(b)** Schematic of the RMF system with one workstation

Figure 2.16: Elements and layout of the RMF systems

To pick an ordered item with the RMFS, the order is first assigned to one of the workstations. Then the item is assigned to a pod and one robot. The robot then moves from its dwell location to retrieve the pod. At this point, the robot moves without a load and can therefore move underneath the pods, without using the designated travel aisles. Once the robot reaches the desired pod, it moves underneath it, lifts the pod, and transports it to the workstation via the travel aisles. The robot enters the workstation buffer and waits for its turn (see Figure 2.16). The picker takes the requested products and adds them to the customer order bin placed in a different rack. The robot then returns the item pod to a storage location that accounts for the frequency of the requests for the pod. The storage locations are therefore fully dynamic (Wurman et al., 2008; Enright & Wurman, 2011). The layout can be fully adapted both dynamically and automatically to the product and order characteristics.

2.7.2 Literature

The performance of RMF systems has hardly been studied scientifically. The literature on the RMF systems is categorized in three categories: *System Analysis, Design Optimization, and Operations Planning and Control*.

System Analysis: Nigam et al. (2014) developed a closed queuing network model for an RMFS. They estimated order throughput time for single-line orders in an RMFS with a turnover class-based storage policy. Lamballais et al. (2017b) extend the work of Nigam et al. (2014) by deriving travel time expressions for multi-line as well as single-line orders in a RMF system with storage zones. They develop a SOQN to estimate the average order cycle time and the utilization of robots and workstations.

Design Optimization: Lamballais et al. (2017b) show that the maximum throughput capacity in a RMF system with storage zones is insensitive to the length-to-width ratio of the storage area (unless the ratio is strongly skewed). However, they show that the positions of the workstations around the storage area directly affects the throughput capacity. Within their settings, the workstations should be located west and east of the storage area when turnover-based zoned storage is used, and north and south of the storage area when zoned storage is not used, to maximize the throughput. Yuan & Gong (2017) develop an OQN to estimate the total throughput time of the RMFS. Using the developed model, they calculate the optimal number of the robots and their required average speed to achieve a certain throughput time. Zou et al. (2018b) optimize the shape of the system using an SOQN.

Operations Planning and Control: Nigam et al. (2014) show that the closest-open location pod storage strategy does not use the storage space efficiently compared to the random location pod storage policy. However, the closest-open location policy achieves a slightly higher throughput capacity. Yuan et al. (2019) investigate the performance of a velocity-based storage assignment for an RMF system. A velocity-based policy is a policy in which the popular items are stored closer the pick stations. By using a fluid model, they show that a 2 or a 3-class velocity-based storage policy reduces the travel distance by 8% and 10%, respectively, in comparison to a random storage policy. Lamballais et al. (2017b) show that the maximum throughput in a RMF system with storage zones can be increased by almost 50% by using pod turnover-based storage zones. One of the drawbacks of Lamballais et al. (2017b)'s analysis is that they assume items on one pod are all the same; for a multi-line order, multiple pods are required. However, in reality, each pod contains multiple products. Therefore, it might be possible that a single pod can fulfill multiple requests of an order. Lamballais et al. (2017c) address this issue by investigating how the inventory of products should be spread across storage pods. They develop an SOQN to estimate the throughput time. They then optimize the number of pods per product, the ratio of the number of workstations to replenishment stations, and the replenishment level for each pod to minimize the throughput time. The results show that the inventory should be spread across as many pods as possible to minimize the throughput time. Furthermore, they find that the optimal ratio of pick stations to replenishment stations is 2 to 1, and that the optimal replenishment level is about 50%. Boysen et al. (2017) investigate sequencing picking orders at the work stations of an RMFS. They formulate the problem as a mixed-integer program. Their results show that by optimally sequencing the picking orders, the order fulfillment process can be done with half of the fleet size of the robots compared to the first come first serve order sequencing rule. Furthermore, they show that the robot fleet can be further reduced by using the shared storage policy, in which the same SKUs are spread over multiple pods. Zou et al. (2018b) investigate different battery recovery strategies for the RMF systems. They develop an SOQN to model the charging process. They conclude that inductive charging provides the best system throughput time. They also realized that the battery swapping strategy outperforms plug-in charging strategy, but it is more expensive.

Table 2.4 presents an overview of the literature on RMF systems.

Table 2.4: Overview of the literature (8 papers) on Robotic Mobile Fulfillment Systems

Research Category	Article	Research Issue	Methodology
System Analysis	Nigam et al. (2014)	Estimate order throughput time for single-line orders	Closed queuing network
	Lamballais et al. (2017b)	Estimate average order cycle time for multi-line orders	Semi-open queuing network
Design Optimization	Lamballais et al. (2017b)	Optimal length-to-width ratio of storage area, optimal location of workstations	Semi-open queuing network
	Yuan & Gong (2017)	Optimal number of robots and their required average speed	Open queuing network
	Zou et al. (2018b)	Optimal shape of the system	Semi-open queuing network
Operations Planning and Control	Nigam et al. (2014)	Efficient pod storage policy	Closed queuing network
	Yuan et al. (2019)	Velocity-based storage assignment	Fluid model
	Lamballais et al. (2017b)	Efficient zoning policy	Semi-open queuing network
	Lamballais et al. (2017c)	Efficient replenishment policy	Semi-open queuing network
	Boysen et al. (2017)	Sequencing picking orders at the workstation	Mixed-integer nonlinear
	Zou et al. (2018b)	Battery charging and swapping strategies	Semi-open queuing network
	Roy et al. (2019a)	Analyze tradeoffs between dedicated and pooled robot assignment, Analyze tradeoffs between random vs. shortest queue allocation of robots	Closed queuing network

2.8 Directions for Future Research

Our review shows that four major topics require investigation for all systems:

1. *System Analysis*: how does the system perform with respect to important performance measures (such as throughput and throughput time) for a given system system configuration?

2. *Design Optimization*: how can the system be designed to optimize certain performance measures, including the optimal shape of the system, and optimal number and locations of workstations?

3. *Operational Policies:* what is the impact of different operational policies on the system performance, such as the effect of storage policies, efficient robot blocking prevention, and dwell point policies?

4. *System Comparison:* how do different systems compare on performance, space and resource utilization, and operational costs?

Not all these questions have been addressed yet for all systems reviewed. In this section, we first discuss important generic research topics for established automated systems, which are not system specific and require further scientific investigation (Section 2.8.1). Next, we provide several research topics that are specific to the automated systems reviewed in this paper, i.e., shuttle systems and RMF systems (Section 2.8.2). We finally identify promising new emerging technologies that have hardly received any research attention yet (Section 2.8.3).

2.8.1 Generic Research Topics for Established Systems

Integrated Models: Almost all existing studies on automated or robotized warehouses analyze storage and pick systems in isolation. For instance, the literature on shuttle systems focuses primarily on storage systems; optimal policies are derived without considering the effect of the storage system configuration on downstream pick performance. Likewise, the literature concerning RMF systems focuses mostly on design issues, rather than operational policies that integrate the picking, storage, and replenishment processes.

To design optimal system configurations, integral consideration of interactions between both upstream processes (such as receiving and reserve storage) and downstream processes (such as picking and packing) in the warehouse is crucial. Such integrated models can capture the variations in the receiving and the picking throughput requirements, which may vary across days and weeks. In particular, the replenishment rates may be more lumpier in comparison to the pick rates. Researchers can take inspiration from integrated models developed in the recent past for container terminal operations (see Meisel & Bierwirth (2013)). While the berth allocation and quay crane allocation problem have been dealt with separately in literature, new algorithms with an integrated focus can improve the joint performance.

Non-Stationary Demand Profiles: Existing research on automated systems focuses largely on performance analysis using stationary inputs. However, due to the ever-changing demand profile, especially in e-commerce environments, it is crucial to take into account non-stationary demand profiles to create a robust design with dynamic operational policies. Sample research questions may be: *How to develop dynamic operational policies, such as selecting dwell point locations and shuttle blocking prevention policies, when the demand profile is non-stationary? What is the optimal shape of the storage area in an RMF system when the demand is non-stationary?* Or, *What is the optimal pod re-positioning policy in an RMF system when the demand changes over time?*

New Storage Policies: Large amounts of e-commerce data provide new insights on customer shopping behavior. In particular, it is possible to estimate, with a very high accuracy, which items will be ordered together. As a result, new storage policies can be introduced that incorporate the product affinity (Mirzaei et al., 2018). The associated research question may be: *How can product affinity be exploited in a storage policy and how does it compare with other storage policies, such as class-based or random?* In decision support systems and marketing literature, market basket analysis (also known as association-rule mining) has been used to discover customer purchase patterns by extracting associations or co-occurrences from transactional databases (see Chen et al. (2005)). Using real-time customer behavior data, dynamic storage policies may be developed, which can improve pick cost and responsiveness.

Product, Order Sequencing: Some of the systems and processes shown in Figure 2.1 (the steps and systems that can be found in fully automated warehouses) have not yet received much research attention. For example, in step four, totes with products have to be retrieved for multiple orders, e.g., from an AVS/R system, to arrive at the stacking robots in the proper stacking sequence (step five and six). Usually these robots have some freedom in item selection. However, it is still very important to have a correct retrieval sequence in order to improve the performance. So the question is: *how can the retrieval shuttles in the AVS/R system be scheduled, with precedence constraints, to improve the stacking process?* Currently, heuristics are used, and much slack is built in the systems. Order sequencing can also improve the efficiency of picking operations for RMF systems. Specifically, by sequencing the orders, it is possible to improve pod coverage, i.e., more items can be picked per pod, see Boysen et al. (2017), who recently published a first paper on this topic.

Multi-Line Order Picking: While single-line orders form the majority of e-commerce order volumes, we expect that the share of multi-line orders will increase, thereby improving packaging efficiencies and reducing carbon footprint. Many retailers offer free shipping for minimum buy quantities. However, most analytical models consider only single line order picking. Obtaining design insights with multi-line order picking is crucial.

Analytical Model Accuracy: Models inherently have to make assumptions to make them tractable. Also, the majority of the existing (stochastic) analytical models have been validated using discrete-event simulation. The credibility of the obtained insights will increase if the models are validated using output measures from real system implementations. In particular, the external queue length measure, which reflects the number of customer transactions waiting to be served, is an important measure for practical decisions. Existing analytical models validated with discrete event simulation reflect errors of up to 40% under various input scenarios (see Roy (2016)). It is not clear how these errors compare to real output data. Other issues, such as non-stationarity of demand, discussed earlier, also play a role. We think more effort must be put in verifying the validity of the output.

2.8.2 Research Topics for Shuttle Systems and RMF Systems

Shuttle Systems

Multiple Input/Output Points: The majority of the literature on shuttle systems provides design and operational choices assuming a single input/output (I/O) point for the system. However, many systems have multiple I/O points. New studies are required to investigate questions such as: *What is the effect of having multiple I/O points on the design and operational choices, such as depth-to-width ratio and storage policies?*

Automated Replenishment: Some systems combine automated storage and replenishment of the pick system (like steps two and three shown in Figure 2.1) with manual picking. Particularly, if the number of pick slots is smaller than the number of products, scheduling the retrievals so that the picker does not have to wait, is challenging. Product bins from which units have been picked already have to be returned to the bulk storage system (step two). This problem has been studied to a limited extent by some researchers, but only in combination with manual pick processes (Yu & De Koster, 2010; Ramtin & Pazour, 2014, 2015; Schwerdfeger & Boysen, 2017; Füßler & Boysen, 2017b,a). Further research is needed on systems with automated picking and for different storage and retrieval configurations.

Robotic Mobile Fulfillment Systems

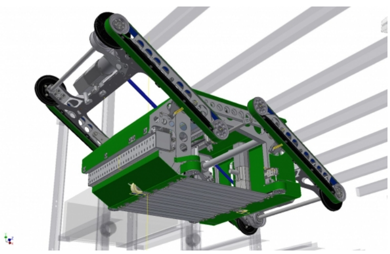
Storage Decisions: For RMF systems, two storage decisions have to be taken. First, how the pods should be stored in the storage area, and second, how the SKUs should be divided over the pods. Artificial intelligence and deep learning can be very helpful to understand order patterns which can be then used to match the right SKUs to pods as well as to decide where to store the pods dynamically every time they have been retrieved for picking.

Replenishment Policy: The pod replenishment policy for RMF systems differs from other systems since multiple SKUs are stored in each pod. Therefore, deciding when to retrieve the pod for replenishment is a challenging question. So the research question is: *What is the optimal inventory threshold for replenishment of the pods?*

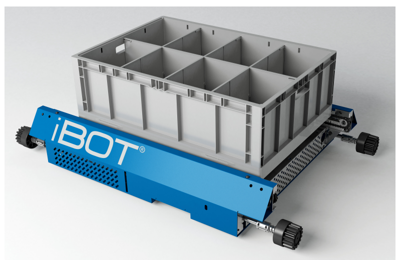
2.8.3 Description and Research Topics for Emerging Technologies

System Description

Vertical and Diagonal AVS/R Systems: In these systems, a single robot can independently roam the storage rack to perform storage and retrieval operations (no lift is required). In a *Diagonal* system, robots also move "diagonally" and in a *Vertical* system robots also move "vertically" inside the rack structure to elevate to the upper levels. The Rack Racer (see Figure 2.17a) developed by Fraunhofer IML is an example of a diagonal system. Perfect Pick® developed by OPEX Corporation and the Skypod™ (see Figure 2.17c) developed by Exotec Solutions are two examples of vertical systems. The Perfect Pick system uses robots, called iBot® (see Figure 2.17b), to perform storage and retrieval actions (Azadeh et al., 2019b).



(a) RackRacer (Source: Fraunhofer)



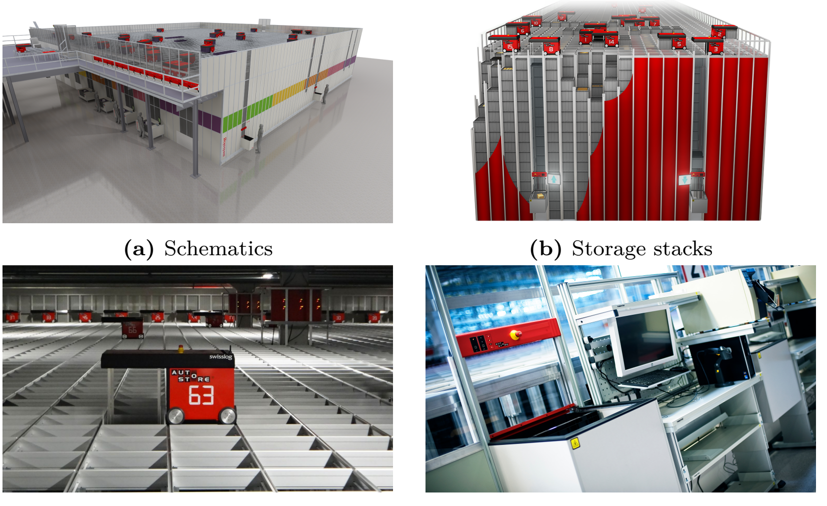
(b) iBot (Source: OPEX)

(c) Skypod™ (Source: EXOTEC)

Figure 2.17: Robots in single-touch systems

The diagonal system has not yet been studied while the vertical system has been studied in only one paper. Azadeh et al. (2019b) models a single aisle of the vertical system using a closed queuing network to optimize the shape of the system. They also investigate the effect of different robot blocking policies on the performance of the system. Finally, they also compare the operational performance and costs of vertical and horizontal systems.

Robot-Based Compact Storage and Retrieval (RCSR) Systems: RCSR systems are another type of grid-based systems (see Section 2.6) in which items are stored in a very dense storage stack with a grid on top. In each cell of the grid, bins that contain the items are stacked on top of each other and form the storage stacks. The workstations are located at the lowest level next to the storage stacks. Robots roam on top of the storage block on the grid. The robots have lifting capabilities, and can extract bins from the storage frames and transport them to the workstations (Zou et al., 2018a). AutoStore$^{\mathrm{TM}}$developed by Hatteland is the first implementation of an RCSR system (see AutoStore (2018)). Recently the British retailer Ocado developed a similar system (Ocado, 2017).



(a) Schematics **(b)** Storage stacks

(c) Robot **(d)** Workstation

Figure 2.18: Robot-based compact storage and retrieval system (Source: Hatteland)

Zou et al. (2018a) are the only ones to investigate the RCSR system. They model the system as a semi-open queuing network and compare two storage policies, namely dedicated and shared storage. They show that the dedicated policy results in a shorter throughput time whereas the shared policy has financial benefits due to substantial cost savings in the total storage space. They also optimize the shape of the system and show that the width-to-length ratio is around 2/3 when using random storage stacks, and slightly larger when using zoned storage racks. They also show that immediate reshuffling can improve the dual command throughput time compared to delayed reshuffling.

GridSort: The GridSort system is based on the GridFlow system discussed in Section 2.6. It uses modular four-directional conveyors, called FlexConveyor (Furmans et al., 2010), or AGVs to transport and sort the loads. Recently, Libiao Robotics has developed a different type of 'GridSort' system, used by several parcel carriers in China that use a fleet of hundreds of autonomous AGVs on a grid to sort parcels by destination.

Pick-Support AMRs: Most retail warehouses still use manual order picking systems. Retail stores usually place large replenishment orders at the distribution center. The DC then ships the orders in multiple roll cages or on pallets. Therefore, a single order requires multiple pick tours (trips between pick locations and the depot). Recently, AMR-based pick systems, called Pick Support AMRs (PS-AMRs), have been developed to minimize the picker travel time to fill large orders. In this system, an AMR automatically follows the picker closely and transports the roll cages, so that the picker can drop off the retrieved items. Once the roll cage is full, the AMR is automatically swapped with a new AMR carrying an empty roll cage. The picker can continue the picking route without returning to the depot, and the AMR automatically transports the full roll cage to the depot. AVGPick^{TM} developed by Swisslog and Pick-n-Go^{TM} developed by Kollmorgen are two examples of such a system (see Pick-n-Go (2010)). Locus Robotics has developed another variant of this system. Instead of following the picker, their AMR (called LocusBots^{TM}) automatically goes to the pick location and waits for the picker to arrive. Once the picker puts the item into a customer tote carried by the AMR, the AMR goes to the next location. When the order is complete, the AMR transports the tote to the depot. Some systems automate the whole picking process (see LocusBots (2018)). An example is the TORU^{TM} picking robot. In this variant, the AMR automatically goes to the picking location and picks up the item without any help from the picker. Similar to the previous variants, once the order is complete, the AMR transports the picked items to the depot (see TORU (2017)).

(a) A picker following AMRs

(b) TORU^{TM} (Source: Magazino)

Figure 2.19: Pick-Support AMRs

Research Topics

All the four general research questions mentioned at the beginning of this section should be investigated for these new technologies (i.e., system analysis, design optimization, operational policy, and system comparison). Furthermore, there are unique characteristics that lead to some system specific research questions.

The distinguishing characteristic of vertical/diagonal systems compared to the horizontal systems is the roaming flexibility of the robots. This provides several routing trajectories to perform storage and retrieval actions. Therefore, a key research question is: *what is the appropriate routing trajectory for the robots, considering performance, blocking delays, and operational costs?*

The differentiating factor of RCSR systems compared to other systems is the fact that items are stacked on top of each other. Therefore, it is crucial to take into account reshuffling and congestion effects when analyzing the system. A particular research question could be: *what is a good storage policy in order to minimize reshuffling and maximize throughput of the system?*

GridSort differs fundamentally from conventional conveyor-based sorters and new models will be required to evaluate its performance. For example, in GridSort the movements of the shuttles (that carry the loads) depend on the empty spaces on the grid. So a research question may be: *how can the empty spaces on the grid be exploited to simultaneously move multiple loads in the system efficiently?*

The differentiating characteristic of PS-AMRs is the collaboration between human pickers and AMRs. The parallel movement of pickers and AMRs makes the modeling, analysis, and optimization of this system completely different from manual picking systems or the robotic systems mentioned earlier in this paper. Evaluating the performance of these systems is an interesting stream for future research. The interesting research question is: *how can we coordinate the parallel movement of the pickers and AMRs in order to maximize the throughput?*

Note that the different systems and research questions will require different methods as suggested in Table 2.1.

2.9 Conclusion

This paper presents an overview of the recent trends in automated warehousing, especially the use of robotic technologies to fulfill orders. The advantages of automation are mainly savings in space, savings on labor costs, 24/7 availability (it is not always easy to find unskilled personnel willing to do warehouse work), and savings on other operational costs, such as heating and lighting. Furthermore, robotic technologies provide scalability and throughput flexibility, which is essential in e-commerce environments where the demand variability is high. Automation of storage and order picking requires considerable scale and a long-term vision as the investments can only be earned back in the medium and longer term. Therefore, it is crucial to develop tools to help decision makers find the correct solutions for their warehouse. As a result, studies have been carried out to model and optimize the performance of the various automated systems. We present modeling techniques, as well as corresponding solution approaches in evaluating the performance of the automated systems. We also illustrate how the models are used in long-term and short-term decision-making processes (design, operational control and planning). We describe well-established automated technologies (AS/R, Shuttle-based and AMR-based systems) as well as the literature related to the various design and control problems in these systems, such as optimally shaping the system, the impact of dwell point policies, block prevention protocols, and storage assignment. These systems differ in terms of infrastructural requirements, operational protocols and equipment movement, and although the frameworks are common, models need to be customized to each system's unique characteristics. We also discuss emerging technologies and aspects that have not received enough (or any) attention in the literature. We summarize the unaddressed research questions in established systems and pose research questions for emerging technologies. Human picking in collaboration with AMRs is one of the most recent technologies that is becoming popular in practice due to its simplicity and flexibility, but has not yet been adequately studied. Also, automated replenishment and sequencing, integrated systems, human-machine interaction and warehouse sustainability are areas that require more attention from researchers.

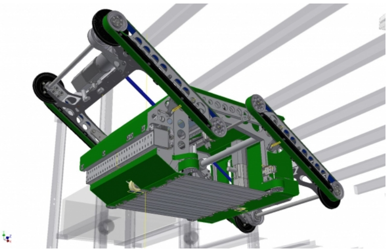
3 Design, Modeling, and Analysis of Vertical Robotic Storage and Retrieval Systems

3.1 Introduction

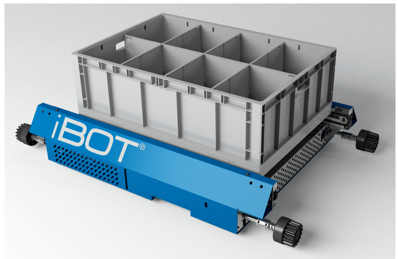
The main challenge in many fulfillment centers is to adapt the picking capacity to the order volume required. The Shuttle- or Autonomous Vehicle-based Storage and Retrieval System (AVS/RS) is one very popular candidate to address this challenge. In this system, a combination of autonomous shuttles and lifts are used to perform the order fulfillment process. Typically, the system throughput is constrained by the number of lifts present in this system. Shuttles move autonomously in the horizontal directions using rails and are transported in the vertical direction using lifts. Hence, we categorize them as *Horizontal* systems.

Recently, robotic-based storage and retrieval systems have been developed that eliminate the multi-touch retrieval process of AVS/R systems. In these systems, a single robot can independently roam throughout the storage rack to transport items between storage locations and workstations. Two variants of robots are used in these systems. In the first group, robots move independently in horizontal and diagonal directions to access a storage location, which we categorize as *Diagonal* systems. One example of the diagonal system is a climbing robot called Rack Racer (see Figure 3.1a) developed by Fraunhofer IML. In the second group, which is the focus of this study, robots move independently and autonomously in the horizontal and vertical directions inside the rack structure. Therefore we categorize them as *Vertical* systems. Perfect Pick® developed by OPEX Corporation is one example of a vertical system, using the iBot® (see Figure 3.1b) as a robot. Skypod™ (see Figure 3.1c) developed by Exotec Solutions is another example of vertical systems. Since 2013, vertical systems have been installed at more than 19 e-retailer warehouses, such as iHerb, BHFO, NewEgg, and Petzl in the U.S., Hudson's Bay in Canada, and Cdiscount in France.

The single-touch retrieval process gives the vertical system an edge over its horizontal counterpart when it comes to flexibility and throughput adjustability. In a given vertical system, the desired throughput level can be obtained by only choosing the correct number of robots. However, in a given horizontal system with an already installed racking structure, the number of shuttles, as well as the number of lifts, need to be adjusted to achieve a certain throughput rate. Furthermore, adding additional lifts requires a major overhaul of the sys-



(a) RackRacer (Source: Fraunhofer)



(b) iBot (Source: OPEX)

(c) Skypod$^{\mathrm{TM}}$ (Source: EXOTEC)

Figure 3.1: Robots in single-touch systems

tem. Moreover, if one of the robots breaks down in the vertical system, it can be replaced without affecting the operation. In contrast, a failure of an exchange point in the horizontal system could result in a system shutdown. Although it seems that the vertical system is more flexible and reliable compared to the horizontal system, it is not clear which system achieves a higher throughput performance with the same storage capacity and number of robots. As a result, a comprehensive study on both systems is required to make that judgment.

Many studies exist that describe and analyze the performance of the horizontal systems (e.g., Malmborg (2002), Marchet et al. (2012), and Roy et al. (2012)), while the vertical system has not been studied yet. Therefore, we first need to investigate the vertical system before we can compare the performance and costs of the two systems. Furthermore, there are fundamental differences between the two systems which leads to a different modeling approach, as well as different layout designs and control policies for the vertical system. In particular, vertical systems have roaming flexibility for robots. This provides multiple routing options to perform storage and retrieval actions, potentially leading to congestion and blocking. This can be mitigated by new block prevention policies (such as a Recirculating policy which we present in Section 3.3.2) that are not possible in horizontal systems. Table 3.1 summarizes some of the key differences between the two systems.

The throughput capacity of the vertical system depends on several design choices, in particular, the layout configuration and the number of operating robots. Although increasing

Table 3.1: Comparison of the features between Horizontal and Vertical systems

Category	Horizontal System (AVS/RS)	Vertical System
Physical Configuration	Shuttles and lifts as storage and retrieval device, up to two L/U points per aisle (each L/U point requires a dedicated lift)	Rack-climbing robots as storage and retrieval device, up to four L/U points per aisle
Vehicle Paths	Rail-based horizontal paths	Flexible horizontal and vertical paths
Load Movement	Loads enter and exit a tier from one head of the aisle	Loads enter the rack section of an aisle from the top and exit from the bottom (or vice versa)
System Throughput	Determined by number of shuttles and lifts.	Determined by number of robots

the number of robots increases the system throughput capacity, it can simultaneously lead to increased blocking delays, potentially reducing the throughput capacity. Therefore, an understanding of the effect of blocking and mitigating policies on performance is crucial, especially in the conceptual design phase. To evaluate the different blocking policies and system designs, one obvious approach is to build detailed simulation models. However, developing a realistic and detailed simulation model for analyzing all possible design scenarios and parameter settings is very time-consuming. Therefore, at the early conceptualization stage, analytical models are used to reduce the design search space and identify a few promising configurations, which can then be fine-tuned using simulation. These analytical models are faster to evaluate, and they allow optimization by enumeration over a large number of design parameters (Tappia et al., 2016; Zaerpour et al., 2015).

Hence, the objective of this paper is to answer the following research questions:

1. How to build accurate and efficient analytical models to analyze the performance of the vertical system?
2. What is an optimal layout for the vertical system in terms of throughput performance?
3. How do the blocking delays affect the throughput performance of the vertical system?
4. Which system is better in terms of costs and throughput capacity: horizontal or vertical?

The focus and prime contribution of this paper lies in the model formulation, analysis, and system comparison. The prime performance metric during the conceptualization phase is the system throughput. We build a closed queuing network to model and use the system and Approximate Mean Value Analysis (AMVA) to estimate the system throughput. By enumerating over the defined search space, we find the optimal shape of the system that maximizes the throughput rate. We analyze a *Recirculation* (REC) blocking policy and compare it with the benchmark *Wait-On-Spot* (WOS) policy for robots that are blocked.

Finally, we compare the performance and operational costs of the vertical and horizontal system.

The rest of the paper is organized as follows. In Section 3.2, the literature review is presented. Section 3.3 describes the vertical system as well as the blocking policies for the vertical system. In Section 3.4, we model the systems as queuing networks and in Section 3.5 we present solution approaches and estimate the performance. In Section 3.6, the numerical analysis is performed. Specifically, we present the optimal configuration of the vertical system and compare the performance of the two blocking policies. Section 3.7 compares the horizontal and the vertical system for costs. Conclusions are given in Section 3.8.

3.2 Literature review

Several studies have developed analytical models to analyze the performance of the horizontal system. The articles can be categorized into three main groups.

Modeling AVS/R Systems: Malmborg (2002) was the first to analyze the horizontal (AVS/R) system. He developed a state equation model to estimate the vehicle utilization and cycle time as a function of system design parameters, e.g., the number of shuttles, lifts, tiers and storage columns. Malmborg (2003b) extended the state equation model by including the pending transactions in the state space description as well to estimate the storage and retrieval cycle time, system utilization and throughput capacity for various system design profiles. Note that the state equation approach is computationally inefficient for solving large-scale problems. Therefore, Kuo et al. (2007) and Fukunari & Malmborg (2008) proposed a nested queuing approach to overcome this problem. In this approach, the lift is modeled as a queuing system in which vehicles are customers and the lifts are servers. The queue is then nested within a separate vehicle queuing system in which transactions are customers and vehicles are servers. The two systems are analyzed iteratively until the performance measures converge. The problem with this approach is its inability to model a scenario in which the cycle starts from outside the storage rack, for example, when loads are received from outside the storage rack. To address this drawback, Fukunari & Malmborg (2009) proposed a queuing network model as an alternative to the nested queuing approach to estimate the performance of AVS/R system. This enabled them to also capture the performance of the system when interfacing with outside systems. Although these models are efficient in estimating the vehicle utilization with reasonable accuracy, they usually fail to estimate the transaction waiting time accurately. Zhang et al. (2009) addressed this problem by dynamically choosing among three different queuing approximations, based on the squared coefficient of variation (SCV) of transaction inter-arrival times. Using this procedure, they were able to increase the accuracy of the transaction waiting time estimation significantly. Heragu et al. (2011) used an open queuing network model and an existing tool called Manufacturing Performance Analyzer (MPA) to analyze the performance of the

AVS/R system. Then they provided an extensive comparison between the performance of AVS/R systems and traditional AS/R systems. Marchet et al. (2012) also used the open queuing network approach to estimate the transaction cycle time of the AVS/R system with tier-captive vehicles. Ekren et al. (2013), Cai et al. (2014), and Roy et al. (2015b) used semi-open queuing networks to evaluate the performance of the AVS/R system with tier-to-tier (pooled) vehicles. Lerher et al. (2015) derived a closed-form expression of the mean travel time for single and dual-command orders in a single-deep AVS/R system. Lerher et al. (2016) developed another closed-form analytical model to estimate the throughput rate of an AVS/R system. Lerher et al. (2017); Ekren et al. (2015) presented a simulation model to calculate single- and dual-command cycle times and throughput of an AVS/R system.

Design Choices for AVS/R Systems: Fukunari et al. (2004) studied the choice of vehicle dwell points in the AVS/R system, using a decision-tree analysis. Kuo et al. (2008) used the queuing network approach to investigate the effect of the class-based storage policy on the cycle time for an AVS/R system. Ekren & Heragu (2010b) used simulation in combination with a regression analysis to analyze the effect of different rack configurations on the system performance. In the regression model, they investigated the effect of three inputs (number of tiers, aisles, and bays) on system performance measures. Ekren et al. (2010) developed a simulation-based experimental design to identify several factors that affect the performance of the AVS/RS. These factors include the L/U point location, dwell point policy, scheduling rule, and interleaving rule and their effect on the storage and retrieval transaction average cycle time, and average utilization of vehicles and lifts. Roy et al. (2012) modeled one tier of the AVS/R system as a multi-class semi-open queuing network with class switching to investigate the impact of design decisions on the performance of the system. They used a decomposition approach to evaluate the effect of different system depth to width ratios, vehicle assignment rules, and multiple storage zones on the expected system cycle time and vehicle utilization. Marchet et al. (2013) presented a design framework for an AVS/R system with a tier-captive shuttles. They used simulation to investigate the effect of rack configuration on the throughput performance of the system. Roy et al. (2015a) used a semi-open queuing network to analyze the optimal choice of dwell-point location and cross-aisle location. Their study showed that the end of the aisle is the optimal location for the cross-aisles, while the L/U point dwell policy improved the performance of the system.

Control Policies for AVS/R Systems: The more complex and stochastic nature of AVS/R system operation requires dynamic and real-time control policies. Especially, in some variants of the system with bi-directional single lanes, there is a possibility of a deadlock in the system. He & Luo (2009) used colored time Petri nets to dynamically model AVS/R systems to establish the necessary conditions to have a deadlock-free system. Roy et al. (2014) investigated the effect of vehicle blocking in the AVS/R system and proposed a semi-open queuing network to analyze the system performance and design trade-offs. Their results showed that the blocking delays could contribute up to 20% of the transaction time.

A detailed simulation model was developed by Roy et al. (2016) to evaluate the blocking delay in the AVS/R system. Carlo & Vis (2012) studied an AVS/R system with two lifts. They proposed a look-ahead strategy heuristic to simultaneously sequence the orders and assign them to each lift.

Zou et al. (2016) investigated a scenario in which the lift and vehicles in the tier-captive AVS/R system are requested simultaneously instead of sequentially. Using a fork-join queuing network, they showed that for a system with less than ten tiers, the parallel processing policy resulted in at least 5% performance improvement compared to the sequential processing policy.

The contribution of this study is threefold:

1. **Modeling:** We are the first to investigate vertical robotic-based storage systems. In these systems, robots move horizontally as well as vertically to perform order transactions. This makes them fundamentally different from previous AVS/R systems. Therefore, new models are required to capture their performance.
2. **Methodological:** We present a jump-over approximation method to analyze the REC policy. This paper is the first to use this technique in a robotic warehousing context.
3. **Design insights:** We are the first to investigate the optimal system size of the new generation robotic-based vertical system. We also compare the effect of two blocking policies on the system throughput capacity. Furthermore, we are the first to compare cost as well as system throughput performance of vertical systems and of the commonly-known horizontal systems (AVS/RS).

3.3 System Description and Assumptions

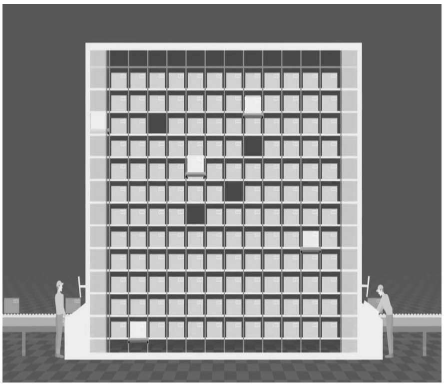
In Section 3.3.1, we describe the vertical system and our modeling assumptions. In Section 3.3.2, we present two blocking policies for the vertical system to investigate the effect of blocking on performance. Table 3.2 presents the main notations used in this study.

3.3.1 Vertical Robotic Storage and Retrieval Systems

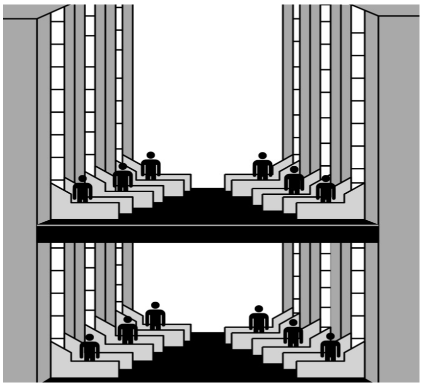
The vertical system consists of several aisles. Each aisle consists of two single-deep storage racks separated by an aisle in which autonomous robots can move. Each robot can access every storage location within the aisle by independently moving horizontally and vertically in sequence. The load and unload point (L/U point) is located at either end or at both ends of each aisle. Additional L/U points can be included on the mezzanine floor to increase the pick capacity (see Figure 3.2). Technically, the robots have the ability to move between aisles. However, in current implementations of the vertical system, the robots are captive to

Table 3.2: Main notations

Notation	Description
N_T	number of tiers
N_C	number of rack sections (columns)
n	total number of storage positions in one aisle of the system ($2N_TN_C$)
h	unit height clearance (tier height)
w	unit width clearance (rack section width)
K	number of robots (vertical) / shuttles (horizontal)
N_{LU}	number of L/U points in the system
v_r	velocity of the robots (vertical system)
v_s	velocity of the shuttles (horizontal system)
v_l	velocity of the lift (horizontal system)
τ_l, τ_u	load and unload time
τ_{LU}	processing time in the L/U point (we refer to this as picking time)



(a) Ground floor L/U point locations (Source: OPEX Corporation)



(b) Mezzanine floor L/U point locations

Figure 3.2: L/U point locations in the vertical system

an aisle. Therefore, in this study, we analyze a single aisle of the vertical system. Figure 3.3 shows a side view of a single aisle of the vertical system with one L/U point. Each aisle is divided into several columns; we denote each column as a rack section. Although there is some level of flexibility for the robots, in recent implementations (see OPEX (2016)), the robot follows a predefined direction path to access each storage location in the aisle. The predefined mostly single directional path minimizes congestion and avoids deadlock. The outer loop is unidirectional while each rack section is bidirectional. Each time an order is placed by a customer, a new retrieval request for a product is made. The control system dispatches the robot from its dwell point to the requested location, following the allowed direction. It picks up the tote containing the ordered product and transports it to the L/U point. A similar process is followed to store a tote back into the storage location. The travel

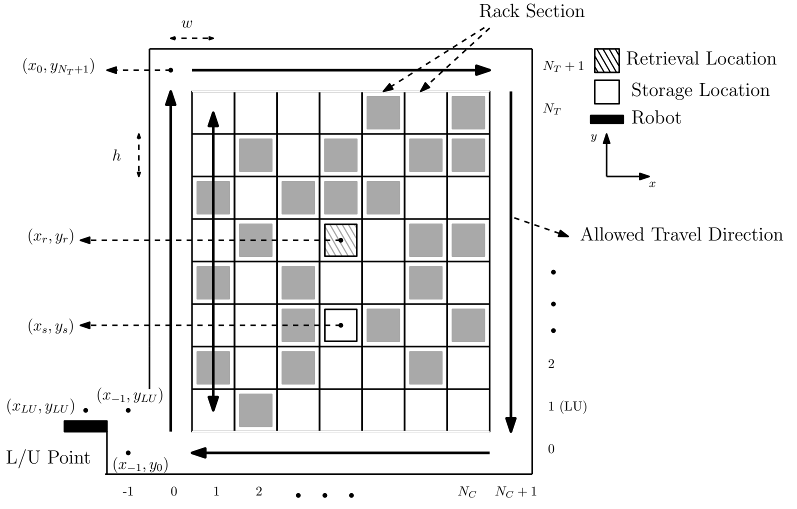


Figure 3.3: Side view of a single aisle of the vertical system

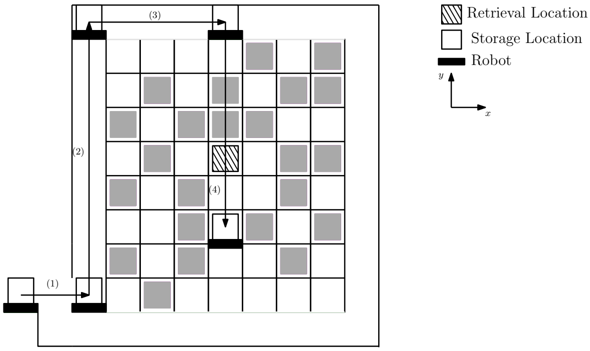
paths corresponding to the processing sequence of the storage and retrieval transactions in a dual-command cycle are depicted in Figure 3.4a and Figure 3.4b respectively.

Travel in the case of two L/U points works as follows (see Figure 3.5). If the dual-command cycle is initiated from the L/U point (a), the robot enters the rack section from the top and exits from the bottom. If the dual-command cycle is initiated from the L/U point (b), the robot enters the rack section from the bottom and exits from the top. Compared to L/U points with shared robots, the travel distance is smaller if the robots are dedicated to an L/U point. The reason lies in the predefined directed paths. If the robot starts from L/U point (a) and wants to go to L/U point (b) after picking up an item, it always passes L/U point (a) before reaching L/U point (b). Consequently, a dedicated policy is used to allocate robots to one of the two L/U points as this minimizes the travel distance. In other words, each robot serves one L/U point.

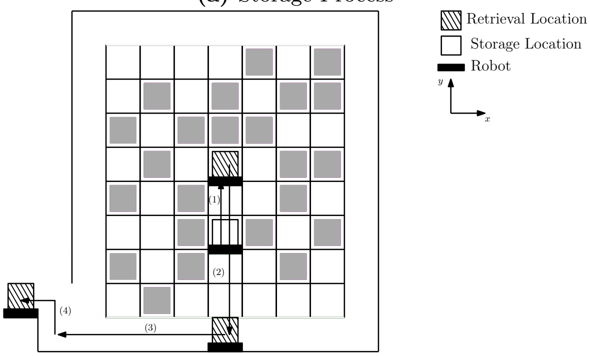
We make the following assumptions in analyzing the vertical system:

1. *Dual-command cycle orders:* We assume the robot first stores the tote containing the previously fetched items (the tote from which the order picker at the L/U point has picked some items), and then retrieves the new tote containing items for the current request. This assumption is in line with the policy that is used in the actual implementation of the system in practice (see OPEX (2016)).

2. *Same rack section:* We assume that the dual-command cycle orders always belong to a specific rack section, i.e., there is always at least one storage location available in



(a) Storage Process



(b) Retrieval Process

Figure 3.4: Dual-command cycle order

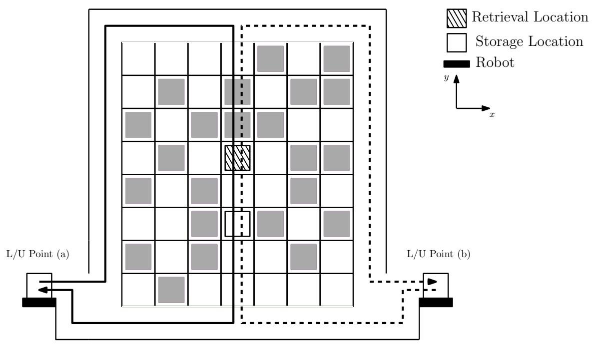


Figure 3.5: Dual-command cycle travel path in case of two L/U points

the same rack section from which the robot will retrieve the new item. Note that this assumption is not very restrictive. An average aisle with N_C rack sections and N_T tiers has $2N_CN_T$ locations. Assume there are p empty locations. With a random storage strategy, the expected number of rack sections without empty locations is

$A = N_C \left(\frac{N_C-1}{N_C}\right)^p$, which approximates to zero for large p and given N_C. Usually, in automated systems, around 10% of the locations in a rack are empty. Therefore, for a typical rack with about 25 rack sections and 25 tiers, the number of empty locations is about $p = 125$, which is large enough to approximate A to zero. Furthermore, storing in the closest open location within the area that has been chosen for retrieval, is a common policy in the robotized systems to minimize the travel time of the robot (Nigam et al., 2014). Also, the actual implementation of the system follows the similar policy most of the time, because it reduces travel distance and increases the throughput capacity of the system. This assumption can be relaxed by routing jobs with a certain probability to other rack sections.

3. *One L/U point:* We assume only one L/U point is available in each aisle unless stated otherwise (the models can be easily extended to accommodate multiple L/U points).

4. *Uniform assignment:* We assume that storage and retrieval locations are assigned uniformly in the system, i.e., the probability of choosing any rack section as well as a location in any rack section is based on the uniform distribution. We further assume that the open locations in the system are independent of the retrieval requests. It is possible to accommodate different storage policies by changing the probability of accessing each rack section.

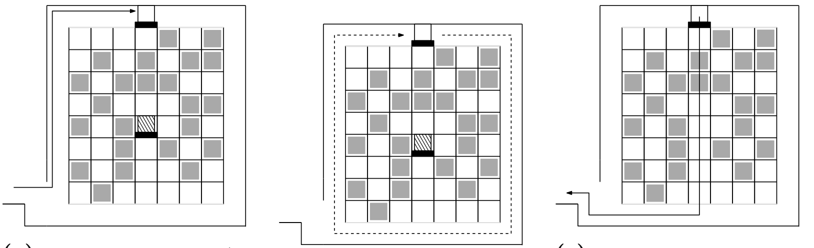
5. *One robot per rack section:* We assume only one robot can access each rack section at a certain time to avoid the risk of a deadlock in the system (because each rack section has a bi-directional path). Note that in the actual implementation of the system similar policy is used most of the time.

6. *Robot velocity:* We incorporate acceleration and deceleration of robots and calculate the average velocity of the robot. We adopt the approach by Lerher et al. (2010) to obtain the average velocity of the robot (see Section 3.9.2). We ignored the driving direction changes because the changeover times are very short. The robots are also assumed to have the same velocity in both horizontal and vertical directions.

3.3.2 Effect of Blocking Delays on the Performance of the Vertical Robotic Storage and Retrieval Systems

To prevent a deadlock in the system, we assume that only one robot is allowed in each rack section at a certain time. If another robot wants to enter an occupied rack section, it needs to wait outside in the outer path, because there is no buffer location in a rack section for the waiting robots to queue. Consequently, increasing the number of robots can potentially lead to congestion and blocking delays in the system. We call the policy in which the blocked robot waits on top of the rack section the *Wait-On-Spot* or WOS policy. It is possible to create limited buffer locations on top of each RS for the blocked robots to queue and ease some blocking congestion. The downside is that a higher rack is required to accommodate the buffers, which increases the robot's travel distance and potentially the system cost.

The drawback of a WOS policy is that the waiting robot might block other robots that want to access another rack section, which negatively impacts the performance of the system. Therefore, we propose another waiting policy to mitigate this problem. In this policy, the *Recirculating* or REC policy, the robot first checks the status of the destination rack section and, if it is occupied, the robot circulates in the outer path around the rack sections. After completing one loop, it checks the status of the rack section again. If the rack section is no longer occupied, the robot claims it. Otherwise, it keeps recirculating until the rack section becomes available (Figure 3.6).



(a) Traveling from L/U point to the top of the rack section

(b) Recirculating while the rack section is occupied

(c) Servicing the rack section and return to L/U point

Figure 3.6: Recirculation policy

The REC policy can potentially result in a lower throughput, especially when the recirculation time is long. Furthermore, because the robots keep recirculating in the system, they consume more energy. However, by using this policy, we are sure that the robots do not block each other while waiting for the rack section to become empty, which can result in a higher throughput for a large number of robots.

3.4 Vertical System Model Description

Increasing the number of robots in the system may cause delays. However, in Section 3.4.1, we first discuss a model where every rack section has unlimited buffer space. This ignores the blocking effect to establish the maximum system throughput capacity. Next, we investigate the effect of blocking delays on the performance of the system. In Section 3.4.2, we present the model for the WOS policy, and in Section 3.4.3, we discuss the model for the REC policy.

3.4.1 Unlimited Buffer Space inside each Rack Section

The process can be divided into three parts (see Figure 3.3): 1) Traveling to the top of the desired rack section: loading the item at the L/U point and transporting it from (x_{LU},y_{LU})

to (x_s, y_{N_T+1}). 2) The process within the rack section: descending from y_{N_T+1} to y_s and storing the tote, then going from y_s to y_r and retrieving the new tote, and finally descending from y_r to y_0. 3) Transporting the item back to the L/U point: going from the bottom of the rack section (x_r, y_0) to the L/U point (x_{LU}, y_{LU}) and unloading the retrieved item tote. The whole process is illustrated in Equation 3.1.

$$\begin{aligned} CT_{dc} =& \tau_{LU} + \frac{|x_{LU} - x_0|}{v_r} + \frac{|y_{LU} - y_{N_T+1}|}{v_r} + \frac{|x_0 - x_s|}{v_r} \\ &+ \frac{|y_{N_T+1} - y_s|}{v_r} + \tau_u + \frac{|y_s - y_r|}{v_r} + \tau_l + \frac{|y_r - y_0|}{v_r} \\ &+ \frac{|x_r - x_0|}{v_r} + \frac{|y_0 - y_{LU}|}{v_r} + \frac{|x_0 - x_{LU}|}{v_r} \end{aligned} \tag{3.1}$$

In this section, we assume that each rack section has sufficient buffer locations for the waiting robots to queue. Figure 3.7 illustrates the corresponding closed queuing network used to estimate the performance of the system.

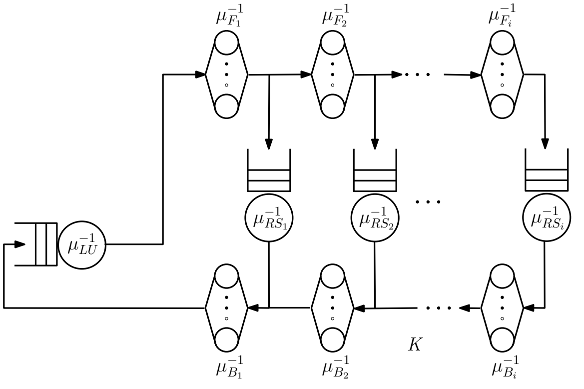


Figure 3.7: Closed queuing network with infinite buffer

Infinite Server (IS) queues with deterministic service times are used to model travel to the top of the rack section (node F_i) and travel from the rack section to the pick station (node B_i). The L/U point is modeled as a single server queue with an exponential service time (node LU) and each rack section is modeled as a single server queue with a generally distributed service time (node RS_i). Although there is only one L/U point in the system, the network can also accommodate a second L/U point by extending it to a multi-class closed queuing network, in which robots are assigned to each L/U point based on their class.

Service Time Expression

Based on the system description and the assumptions, the expected service time of each node can be calculated. μ_{LU}^{-1} depends on the speed of the picker. Equations 3.2 and 3.3 show the derivations of $\mu_{F_i}^{-1}$ and $\mu_{B_i}^{-1}$, respectively in which $i = 1, .., N_C$ corresponds to the rack section number.

$$\mu_{F_i}^{-1} = \begin{cases} \frac{|x_{LU}-x_0|}{v_r} + \frac{|y_{LU}-y_{N_T+1}|}{v_r} + \frac{|x_0-x_1|}{v_r} = \frac{2w}{v_r} + \frac{N_T h}{v_r} + \frac{w}{v_r} & \text{, if } i = 1 \\ \frac{|x_i-x_{i-1}|}{v_r} = \frac{w}{v_r} & \text{, } Otherwise \end{cases} \tag{3.2}$$

$$\mu_{B_i}^{-1} = \begin{cases} \frac{|x_1-x_{-1}|}{v_r} + \frac{|y_0-y_{LU}|}{v_r} + \frac{|x_{-1}-x_{LU}|}{v_r} = \frac{2w}{v_r} + \frac{h}{v_r} + \frac{w}{v_r} & \text{, if } i = 1 \\ \frac{|x_i-x_{i-1}|}{v_r} = \frac{w}{v_r} & \text{, } Otherwise \end{cases} \tag{3.3}$$

Equation 3.4 presents the travel time within RS_i. Note that y_s and y_r, the storage and retrieval locations within the rack section, are uniformly distributed.

$$T_{RS_i} = \frac{|y_{N_T+1} - y_s|}{v_r} + \tau_u + \frac{|y_s - y_r|}{v_r} + \tau_l + \frac{|y_r - y_0|}{v_r} \tag{3.4}$$

The expected value and squared coefficient of variation of the service time of RS_i is:

$$\mu_{RS_i}^{-1} = E[T_{RS_i}] \tag{3.5}$$

$$cv_{RS_i}^2 = \frac{Var[T_{RS_i}]}{[E[T_{RS_i}]]^2} \tag{3.6}$$

3.4.2 No Buffer Space inside a Rack Section and WOS Blocking Policy

In the WOS policy, the robots move to the top of the rack section and check whether the rack section is occupied. If it is not occupied, it moves to the rack section. If not, it waits on top. This process can be modeled using the blocking-after-service (BAS) protocol (Perros, 1994), in which the robot either goes to the rack section node or waits at the previous node if the rack section is full. Figure 3.8 presents the corresponding queuing network.

In this network, all the nodes are modeled as single-server queues with exponentially distributed service times (this is needed for the solution approach to solve this network, which will be discussed in Section 3.5.2). The L/U point is assumed to have unlimited buffer space, while all the other travel nodes do not have any buffer space. The rack sections can also have a limited number of robot buffer locations. Using an unlimited buffer at the L/U point

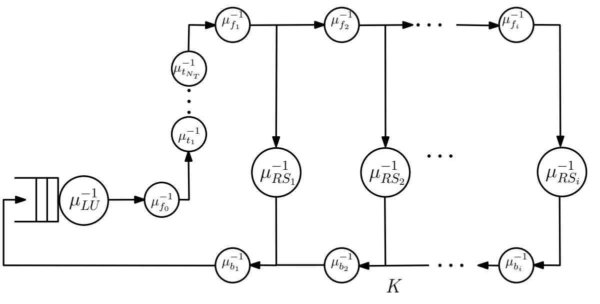


Figure 3.8: Closed queuing network with finite buffer (WOS policy)

is justified because in practice the robots can wait on a conveyor with ample space. The description of each node, as well as its expected service time, is presented in the next part.

Service Times Expressions

The expected service time in the L/U point and in the rack section do not change when including blocking in the system, therefore μ_{LU}^{-1} and $\mu_{RS_i}^{-1}$ have the same value as the unlimited buffer network presented in Section 3.4.1. $\mu_{f_0}^{-1}$ is the travel time to move horizontally from the L/U point to the location (x_0, y_1) in Figure 3.3, with expectation $\frac{2w}{v_r}$. $\mu_{t_i}^{-1}$ is the travel time to climb one tier, with expectation $\frac{h}{v_r}$. Similarly, $\mu_{f_i}^{-1}$ and $\mu_{b_i}^{-1}$ are the travel times to move horizontally the width of one RS, with expectation $\frac{w}{v_r}$; note that $i = 1, .., N_C$.

3.4.3 No Buffer Space inside a Rack Section and REC Blocking Policy

First, we build a network for one rack section and then we extend it to the whole system. Upon service completion, the robot goes from the workstation to the top of the rack section. This process is defined by two infinite server queues F and UP. If the rack section is not occupied, the robot enters the rack section. Otherwise, it recirculates in the system by going to the infinite server queue D which in combination with node UP creates the outer recirculation loop. Note that the robot recirculates in the outer loop of the system (Figure 3.6b). After completing one loop, the robot checks the rack section again. Let p_b be the probability that the rack section is occupied (p_b is the same as the marginal probability of having one robot in RS). Then, with probability p_b, the robot goes to node D and with probability $1 - p_b$, the robot enters node RS. Upon service completion in RS, the robot continues its route to the L/U point via the infinite server queue G, which represents the travel time between the rack section and the L/U point. Figure 3.9 illustrates the resulting closed queuing network.

Because the waiting robots circulate in the outer loop without stopping anywhere, we assume that no blocking occurs in the outer loop. Therefore, we can treat each rack section

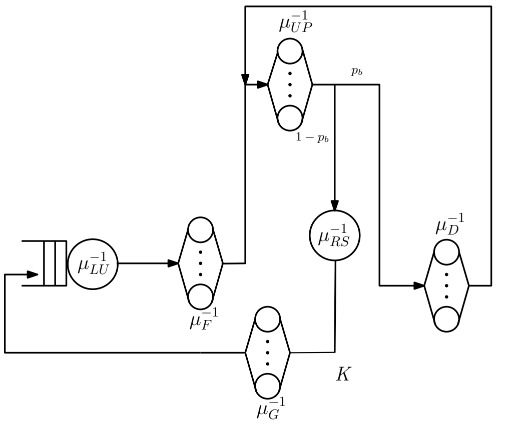


Figure 3.9: Closed queuing network for one RS

separately and extend the previous network to accommodate the rest of the rack sections. Figure 3.10 presents the resulting network.

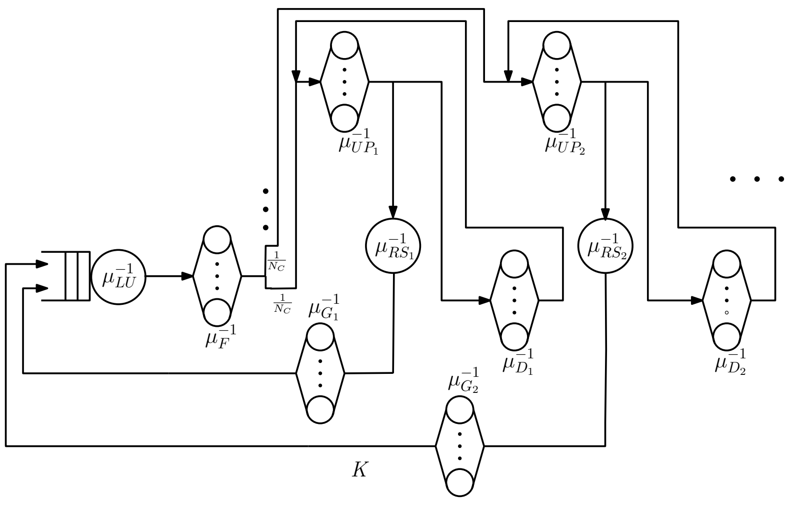


Figure 3.10: Closed queuing network with finite buffer (REC policy)

Service Times Expressions

The terms, $\mu_{RS_i}^{-1}$ and μ_{LU}^{-1} have the same value as in Section 3.4.1. μ_F^{-1} is the travel time to move horizontally from the L/U point to the location (x_0, y_1) in Figure 3.3. $\mu_{UP_i}^{-1}$ is the travel time to go from (x_0, y_1) to the top of RS_i, and $\mu_{B_i}^{-1}$ is the travel time to go from bottom of the RS_i to the L/U point. $\mu_{D_i}^{-1}$ is the travel time to go from the top of the RS_i to (x_0, y_1), so that $\mu_{D_i}^{-1}$ plus $\mu_{UP_i}^{-1}$ creates the whole loop time. μ_F^{-1}, $\mu_{UP_i}^{-1}$, $\mu_{B_i}^{-1}$, and $\mu_{D_i}^{-1}$ are presented in Equations 3.7, 3.8, 3.9, and 3.10, respectively; note that $i = 1, .., N_C$.

$$\mu_F^{-1} = \frac{2w}{v_r} \tag{3.7}$$

$$\mu_{UP_i}^{-1} = \frac{N_T h}{v_r} + \frac{(i)w}{v_r} \tag{3.8}$$

$$\mu_{G_i}^{-1} = \frac{iw}{v_r} + \frac{h}{v_r} + \frac{2w}{v_r} \tag{3.9}$$

$$\mu_{D_i}^{-1} = \frac{((N_C + 1) - i)w}{v_r} + \frac{(N_T + 1)h}{v_r} + \frac{(N_C + 1)w}{v_r} + \frac{h}{v_r} \tag{3.10}$$

3.5 Solution Approach

In this section, we provide three solution approaches, to solve the networks that are presented in Section 3.4. Section 3.5.1 offers a solution to solve the network with unlimited buffer locations. Section 3.5.2 discusses the solution approach for solving the network for the WOS policy, and Section 3.5.3 presents a solution algorithm to solve the network corresponding to the REC policy.

3.5.1 Closed Queuing Network with Unlimited Buffer

The RS_i nodes corresponding to the closed queuing network described in Section 3.4.1 have generally distributed service times. Therefore, the networks do not have a product form solution (Baskett et al., 1975) and an approximation technique is required. We use Approximate Mean Value Analysis (AMVA) to approximate the performance of the networks. The AMVA method is an extension of the regular MVA method (Reiser & Lavenberg, 1980), in which the squared coefficient of variation of the service time is used as a correction factor in calculating the residence times. The method is explained in Section 3.9.1.

Throughput Validation with Real System

To validate the model for the vertical system, we compare the result with the output data provided by Bastian Solutions, a consulting company that has implemented such systems in practice. Table 3.3 presents the system parameters.

The company only reported the maximum velocity of robots without accounting for acceleration/deceleration, to be 1.89 m/s. We use the approach by Lerher et al. (2010) to calculate the average velocity of the robot (see Section 3.9.2). We assume that the robot

Table 3.3: System parameters (Bastian Solutions, 2016)

$n/2$	N_T	N_C	w	h	K	v_r	τ_l, τ_u	τ_{LU}
875	25	35	80 cm	32 cm	15	1 m/s	1.5 sec	5 sec

has an acceleration/deceleration of 1 m/s^2 and calculate the average velocity of the robot inside the rack section. The resulting average velocity of the robot is 1 m/s which we use in the rest of this study. To estimate w and h, we have added 5% to the tote dimensions as reported by the company, to account for the rack size and other clearances in the rack structure. The system also has two L/U points, one at each end of the aisle. Using the given parameters, the system achieves a throughput of 800 dual-command cycles per hour according to the company. Using our analytical model, we estimate the system throughput to be 824 dual-command cycles per hour (about 3% higher than what is achieved). We also build a simulation model using the parameters in Table 3.3 and compare the results with our analytical model performance. The errors, which are reported in Table 3.4, are negligible.

Table 3.4: Performance statistics

	Sim	Analytical	Error
Utilization Rack Section 1	0.5708 ± 0.0146	0.5674	0.6%
Utilization L/U Point	0.1714 ± 0.016	0.1698	0.93%
System Throughput (DC/h)	822.8346 ± 1.2519	824	0.14%

DC/h: dual-command cycle per hour

3.5.2 Closed Queuing Network without Buffer Location - WOS Policy

Due to the finite buffers, the queuing network developed in Section 3.4.2 does not have a product form solution (Perros, 1994) and an approximation is needed to obtain the performance statistics. We use the approximation proposed by Akyildiz (1988) because it fits our network's characteristics. The approximation is a modified MVA for calculating the mean residence time of the jobs in the blocked node. First, the number of jobs in the destination node is calculated using MVA. If the calculated number of jobs is bigger than the number of buffer spaces in the destination node, the new job is blocked. Therefore, the new job remains in the source node until a spot in the destination node becomes available. When a job finishes service at the destination node, the spot becomes available for the blocked job. Consequently, the expected residence time of the new jobs in the source node increases by the expected remaining service time of the job in the destination node. The expected remaining service time of the job in the destination node is approximated to be the same as the expected service time of the destination node. The detailed explanation of the algorithm and its accuracy evaluation can be found in Akyildiz (1988).

3.5.3 Closed Queuing Network without Buffer Location - REC Policy

The queuing network developed in Section 3.4.3 does not have a product form solution either. To accurately estimate the performance statistics, we approximate the network by another one with the jump-over blocking protocol (Van Dijk, 1988). The jump-over network has a product form solution, and its performance statistics can be calculated using MVA (Van der Gaast et al., 2020). We first present how the jump-over network is built, and then we illustrate the MVA based algorithm to solve the network. Table 3.5 presents the notations used in explaining the jump-over network and the solution algorithm.

Table 3.5: Notations for jump-over network

Notation	*Description*
k	number of robots in the network
K	total number of robots in the system
j, m, n	nodes in the network: LU, F, UP_i, RS_i, D_i, B_i
i	rack section index: 1, 2, ..., N_C
$T_j(k)$	mean residence time of node j when there are k jobs in the network
$L_j(k)$	mean number of robots at node j when there are k jobs in the network
$X(k)$	overall throughput of the system when there are k jobs in the network
$P_{m,n}$	routing probability of going from node m to node n
p_{b_i}	blocking probability of node RS_i
$U_j(k)$	utilization of node j when there are k jobs in the network
$\pi_j(q\|k)$	marginal probability of having q jobs in node j when there are k jobs in the system
V_j	visit ratio for node j
μ_j^{-1}	expected service time of node j

Jump-over Network

Assume a robot that intends to visit RS_i is labeled as either *entered* RS_i or *skipped* RS_i. Technically, this labeling is done based on the robot actually visiting the RS_i, i.e., it is labeled *entered* RS_i if it actually entered RS_i and served, and it is labeled *skipped* RS_i if it skipped RS_i because the rack section was occupied. Alternatively, we can say that the robot always enters RS_i but incurs zero service time if it is labeled *skipped*. However, in the jump-over network, the labeling is done randomly and regardless of whether the robot actually entered RS_i and received service. The probability of a robot receiving either one of the labels is taken as the fraction of robots in the original network receiving the specific label, i.e., the fraction of robots that are labeled as *skipped* RS_i equals the blocking probability of RS_i in the original network (p_{b_i}). Initially, p_{b_i} is not known; however, it can be estimated iteratively using any initial guess. Hence, it is assumed that p_{b_i} is known beforehand in the jump-over network. Therefore, P_{RS_i,D_i} (routing probability to go from RS_i to D_i) equals p_{b_i} and P_{RS_i,G_i} (routing probability to go from RS_i to G_i) equals to $1 - p_{b_i}$. This means that the robot is labeled *skipped* RS_i and routed to the node D_i with probability p_{b_i}, and it is labeled *entered* RS_i and routed to the node G_i with probability $1 - p_{b_i}$. Finally, after the robot comes out of the RS_i, it follows the mentioned Markovian route regardless of whether it skipped or served inside RS_i. Figure 3.11 presents the corresponding network with the jump over protocol.

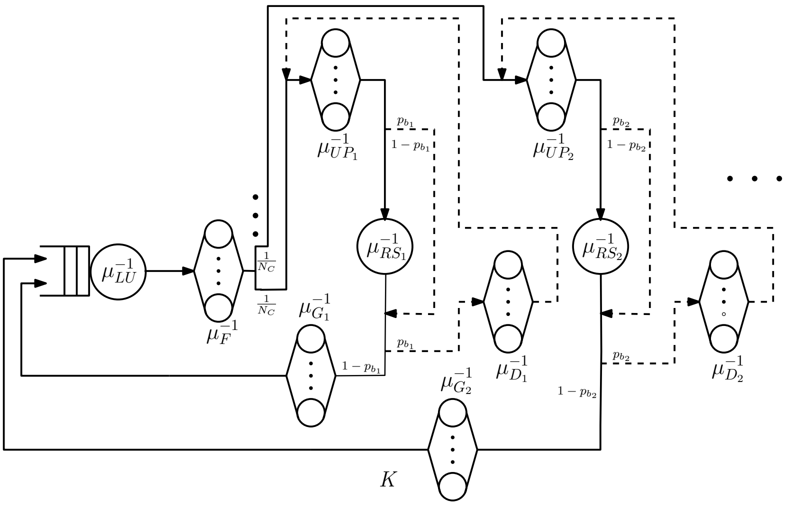


Figure 3.11: Jump-over approximation of the closed queuing network in Figure 3.10

Visit Ratios

The visit ratios (V_j) can be calculated for each node according to the routing probabilities. We illustrate the approach to obtain visit ratios with one rack section in the system. Then, using the same idea, the visit ratio of the system with more than one rack section can be obtained. For that purpose, we follow a robot doing a dual-command cycle route, starting from the L/U point and coming back to the L/U point. Depending on how many times the robot has jumped the rack section, the number of times it visited each node in one route is different. Table 3.6 presents the number of times the robot visits each node on its route for different scenarios, along with the probability of a scenario.

The number of times the robot visits LU, F and G in a dual-command cycle is always one. Therefore, the expected number of visits for LU, F and G is one ($V_{LU} = 1$, $V_F = 1$, $V_G = 1$). Note that the robot always visits a RS even if it is blocked, but the incurred service time is zero. It is clear from the Table 3.6, that the number of times the robot visits UP or RS follows a geometric distribution with parameter $1 - p_b$. Therefore, the expected number of

Table 3.6: Number of times the robot visits each node

LU	F	UP	RS	D	G	Probability
1	1	1	1	0	1	$1 - p_b$
1	1	2	2	1	1	$(1 - p_b)p_b$
1	1	3	3	2	1	$(1 - p_b)p_b^2$
1	1	4	4	3	1	$(1 - p_b)p_b^3$
.	.	.	.	.	.	.
.	.	.	.	.	.	.

visits for UP and RS is $V_{UP} = V_{RS} = \frac{1}{1-p_b}$. The number of times the robot visits D also follows the same geometric distribution but starting from zero. Hence, the visit ratio for D is $V_D = \frac{p_b}{1-p_b}$. Similarly, the visit ratios for a network with more than one rack section can be obtained by multiplying them with the probability that the RS_i is selected.

Mean Value Analysis for Jump-over Network

The idea is to initialize the p_{b_i} with any arbitrary value, then to solve the jump-over network using MVA, update the blocking probabilities and repeat the MVA again until the difference between the current calculated blocking probability and the previous one is less than a given ϵ.

In each iteration of the MVA, given k robots in the system, the algorithm calculates $T_j(k)$, $X(k)$, $L_j(k)$, and $\pi_{RS_i}(1|k)$ (the marginal probabilities that RS_i is occupied). By iteratively increasing the number of robots in the system from zero to K, we can calculate all the performance statistics. For the initialization phase ($k = 0$), we have $L_i(0) = 0$, $\pi_{RS_i}(0|0) = 1$, $\pi_{RS_i}(1|0) = 0$. Using the arrival theorem, the mean residence time of the LU, F, UP_i, D_i, and G_i are calculated as:

$$T_j(k) = \begin{cases} \frac{1}{\mu_j}(1 + L_j(k-1)) & \text{if } j = LU \\ \frac{1}{\mu_j} & \text{if } j = F, UP_i, D_i, G_i \end{cases} \tag{3.11}$$

The mean residence time of the RS_i is calculated by the following expression:

$$T_{RS_i}(k) = p_{b_i}(0) + (1 - p_{b_i})\frac{1}{\mu_{RS_i}} \tag{3.12}$$

p_{b_i} is the probability that the RS_i is occupied, i.e., the probability of having one robot in RS_i when we have $k-1$ robots in the system.

$$p_{b_i} = \pi_{RS_i}(1|k-1) \tag{3.13}$$

as a result:

$$T_{RS_i}(k) = \pi_{RS_i}(1|k-1).0 + (1 - \pi_{RS_i}(1|k-1))\frac{1}{\mu_{RS_i}} \tag{3.14}$$

Once the $T_j(k)$s are obtained, the system throughput $X(k)$ is calculated by using Little's Law:

$$X(k) = \frac{k}{\sum_j V_j T_j(k)} \tag{3.15}$$

Using the throughput of the system and using Little's Law again, the new number of robots in each node for the next iteration is calculated:

$$L_j(k) = V_j X(k) T_j(k)$$

The marginal probabilities of having q robot in RS_i are calculated by balancing the number of transactions per time unit between state $q-1$ and q. The rate from q to $q-1$ is given by $\mu_{RS_i}\pi_{RS_i}(q|k)$, and by using the arrival theorem, the rate of going from $q-1$ to q is $V_{RS_i}X(k)\pi_{RS_i}(q-1|k-1)$. Because q is either one or zero for RS_i we have:

$$\pi_{RS_i}(1|k) = \frac{V_{RS_i}X(k)}{\mu_{RS_i}}\pi_{RS_i}(0|k-1) \tag{3.16}$$

Furthermore, from the normalization constraint we have:

$$\pi_{RS_i}(0|k) = 1 - \pi_{RS_i}(1|k) \tag{3.17}$$

Therefore, the marginal probabilities can be calculated by exploiting Equations 3.16 and 3.17.

The performance statistics are estimated by sequentially applying the above equations. However, before the procedure starts, the blocking probability needs to be initialized. The initial value for the blocking probabilities can be any arbitrary number between zero and one. Based on this initial value for the blocking probabilities and by using the MVA method, the marginal probability of finding RS_i containing one robot is obtained. Then, we use this value as a new initial value for the blocking probabilities and repeat the MVA again.

$$p_{b_i}^{new} = \pi_{RS_i}(1|K) \tag{3.18}$$

We continue the procedure until the difference between the newly calculated blocking probabilities and the current one is less than an ϵ.

$$|p_{b_i}^{new} - p_{b_i}^{current}| < \epsilon$$

In our experiments, convergence is reached fast (within a second) regardless of the initial values of p_{b_i}. The total procedure is presented in Section 3.9.3.

Validation

We use simulation to see whether the jump-over approximation accurately estimates the performance statistics of the system with the REC policy. Instead of simulating the queuing network, we simulate a realistic implementation of an actual system to validate the model. The detailed simulation is built using the parameters listed in Table 3.7.

Table 3.7: System parameters

$n/2$	N_T	N_C	w	h	K	v_r	τ_l, τ_u	τ_{LU}
8	4	2	80 cm	32 cm	15	1 m/s	1.5 sec	3.6 sec

The simulation is built in AutoMod^{TM}software version 12.3, simulation software commonly used by many material handling companies. The model uses the actual REC blocking policy and the exact physical dimensions of racks, totes, and robots, and exact travel times. Only the picking process is assumed to have an exponential distribution with unlimited buffer capacity. We run the simulation 40 times and record the results with 90% confidence intervals. Then, using the same parameters, we create the jump-over approximation of the system and solve it using the proposed algorithm. The blocking probabilities of each rack section and the system throughput of the system obtained by simulation and the analytical model are compared with two levels for the number of robots K in the system: five and 10. Table 3.8 illustrates the results.

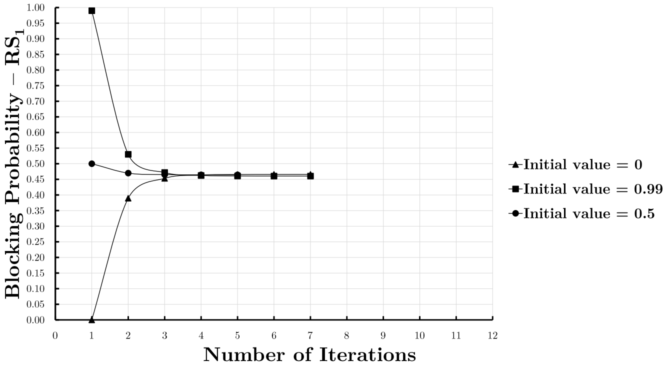
Table 3.8: Performance statistics

K	Blocking Probability - RS_1			Blocking Probability - RS_2			System Throughput (DC/h)		
	Sim	Jump	Error	Sim	Jump	Error	Sim	Jump	Error
5	0.4791 ± 0.0091	0.4626	3.44%	0.4535 ± 0.0132	0.4657	2.69%	683.12 ± 4.30	687.49	0.64%
10	0.6393 ± 0.0081	0.6543	2.34%	0.6787 ± 0.0053	0.6582	3.02%	866.51 ± 6.41	895.98	3.40%

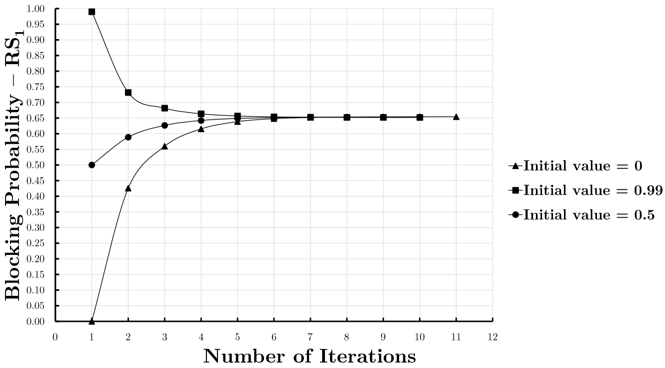
DC/h: dual-command cycle per hour

The jump-over approximation results in less than 4% error, and is, therefore, an accurate representation of the original system with the REC blocking policy.

Furthermore, Figure 3.12 illustrates the number of iterations required by the algorithm to converge for different initial values of p_{b_1}. The ϵ is set to 0.0001.



(a) $K = 5$



(b) $K = 10$

Figure 3.12: Number of iterations the algorithm needs to converge

3.6 Numerical Analysis

Choosing the right dimensions of the system directly influences the performance of the system. In Section 3.6.1, we investigate different layout configurations of the vertical system using the model developed in Section 3.4.1, to obtain the optimal layout. Later in Section 3.6.2, we compare the performance of the two blocking policies modeled in Section 3.4.2 and Section 3.4.3 to evaluate under which conditions the blocking policies result in better system performance.

3.6.1 Optimal Rack Layout Configuration

Using the closed queuing network developed in Section 3.4.1 (i.e., no blocking is assumed), we analyze different layout configurations, with varying height to width ratio (i.e., N_T/N_C), to obtain the optimal layout configuration. We use discrete optimization where we enumerate over possible parameter values in the design search space to find the optimal value. We formulate the optimization problem as follows.

$$\begin{aligned}
\underset{N_T,N_C}{\arg\max} \quad & X(n,K,N_T,N_C) \\
\text{s.t:} \quad & \\
& \begin{array}{rcl}
N_T N_S & = & \frac{n}{2} \\
LB_{N_T} & \leq & N_T \\
N_T & \leq & UB_{N_T} \\
LB_{N_C} & \leq & N_C \\
N_C & \leq & UB_{N_C} \\
K & = & constant \\
\frac{n}{2} & = & constant \\
N_T, N_C & \in & \mathbb{Z}^+.
\end{array}
\end{aligned} \tag{3.19}$$

We investigate four levels of the number of storage locations ($n/2 = 300$, 600, 900, 1200) and two levels of the number of robots in the system ($K = 5$ and 10). The rest of the system parameters are the same as shown in Table 3.3. In each scenario, the system throughput is calculated while changing the number of tiers and rack sections for their Lower Bound (LB) to their Upper Bound (UB) in a way that the total number of storage locations remains the same. The layout resulting in the maximum system throughput is identified as the optimal layout. Figure 3.13 illustrates system throughput for different scenarios. The $x-axis$ is the value of $\frac{N_T}{N_C}$ in a logarithmic scale and the $y-axis$ is the system throughput, measured in the number of dual-command cycles per hour.

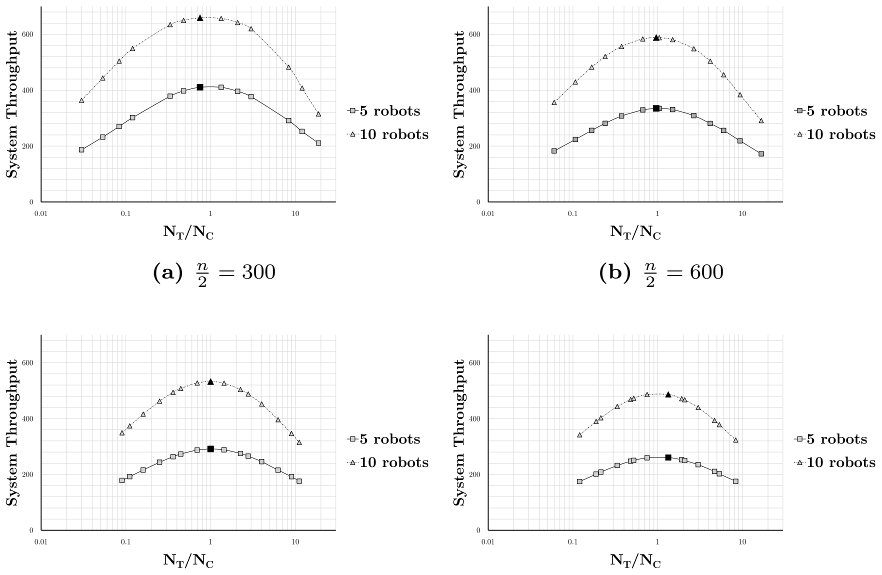
Table 3.9 presents the layout configuration which achieves the maximum system throughput.

Table 3.9: Optimal layout configurations

$n/2$	R	N_T	N_C	$(N_T/N_C)^*$	Throughput (DC/h)
300	5	15	20	0.75	410.54
300	10	15	20	0.75	659.07
600	5	24	25	0.96	334.94
600	10	24	25	0.96	588.35
900	5	30	30	1.00	291.00
900	10	30	30	1.00	532.21
1200	5	40	30	1.33	260.45
1200	10	40	30	1.33	486.31

DC/h: dual-command cycles per hour

Because N_T and N_C take integer values, only certain ratios of N_T/N_C can be realized for a given number of storage locations. As a result, the presented optimal N_T/N_C ratios are not



(a) $\frac{n}{2} = 300$ (b) $\frac{n}{2} = 600$

(c) $\frac{n}{2} = 900$ (d) $\frac{n}{2} = 1200$

Figure 3.13: System throughput in different scenarios

identical for layouts with a different number of total storage locations. However, Table 3.9 and Figure 3.13 show that the optimal N_T/N_C ratios are close to one in each scenario.

Instead of using a queuing network, some studies adopt a probabilistic travel time approach to obtain the optimal layout configuration. In that method, the system configuration is chosen to minimize the expected dual-command travel time. In Section 3.9.5, this approach is presented for the vertical system. The results are presented in Table 3.10. The reported numbers are the feasible values for N_T and N_C while the numbers in parentheses are the actual numbers calculated by the method.

Analysis and obtained insights: Bozer & White (1984) show that for a random storage assignment, the square-in-time rack is optimal for an automated storage and retrieval system, i.e., the expected horizontal travel time is equal to the expected vertical travel time in a single- or dual-command cycle. Our travel time analysis also provides similar results for the vertical system. Furthermore, the optimal layout resulting from the travel time expression approach leads to almost the same system throughput as the queuing network. Hence, in this particular case, we can claim that the waiting times in the L/U point and rack section queues in the network do not affect the optimal layout configuration.

Table 3.10: Optimal layout configurations using dual-command throughput time expression approach

$n/2$	R	N_T	N_C	N_T/N_C	Throughput (DC/h)
300	5	20 (17.92)	15 (16.74)	1.33 (1.07)	410.36
300	10	20 (17.92)	15 (16.74)	1.33 (1.07)	656.83
600	5	25 (25.35)	24 (23.66)	1.04 (1.07)	334.80
600	10	25 (25.35)	24 (23.66)	1.04 (1.07)	588.05
900	5	30 (31.05)	30 (28.99)	1.00 (1.07)	291.00
900	10	30 (31.05)	30 (28.99)	1.00 (1.07)	532.21
1200	5	40 (35.86)	30 (33.46)	1.33 (1.07)	260.45
1200	10	40 (35.86)	30 (33.46)	1.33 (1.07)	486.31

DC/h: dual-command cycle per hour

3.6.2 Comparing the Blocking Policies

Two blocking policies were modeled in Section 3.4.2 and Section 3.4.3. In this section, we aim to find out how the REC policy performs compared to the basic WOS policy. We investigate the WOS policy when there is no buffer location as well as when there is one buffer location inside each RS (WSO-0 and WSO-1 respectively). We analyze the performance of a system with four rack sections and eight tiers ($N_C = 4$, $N_T = 8$). For the WOS-1 policy, the system has nine tiers ($N_T = 9$) to accommodate the buffer location. In order to see the blocking effect on performance, the system needs to be the bottleneck and not the picker. Therefore, we assume to have four servers (pickers) at the single L/U point each with a picking time of 3.6 sec (which corresponds to the current state-of-the-art picking stations that achieve 1000 dual-command cycles per picker per hour). The high picking speed at the L/U point prevents the picking process become the bottleneck. The remaining system parameters are the same as in Table 3.3. In this scenario, we increase the number of robots in the system from one to 15 and observe the system throughput obtained by the two blocking policies. The results are illustrated in Figure 3.14. The $x - axis$ represents the number of robots in the system and the $y - axis$ is the system throughput, measured in dual-command cycles per hour.

Analysis and obtained insights: The graph (Figure 3.14) shows that the policies perform approximately the same with a small number of robots in the system (the WOS-0 policy has a slight advantage). However, if we increase the number of robots, the system throughput decreases sharply under the WOS policy. This decrease can be attributed to the waiting robots that block passing robots in the outer loop. The reason for the throughput drop is as follows: there are some waiting positions in the system depending on the number of tiers and rack sections (see Figure 3.8), and once all the waiting positions are occupied, the blocking delays cause the system throughput to drop. In our example, the throughput using WOS-0 policy drops if we increase the number of robots from eight to nine because this can result

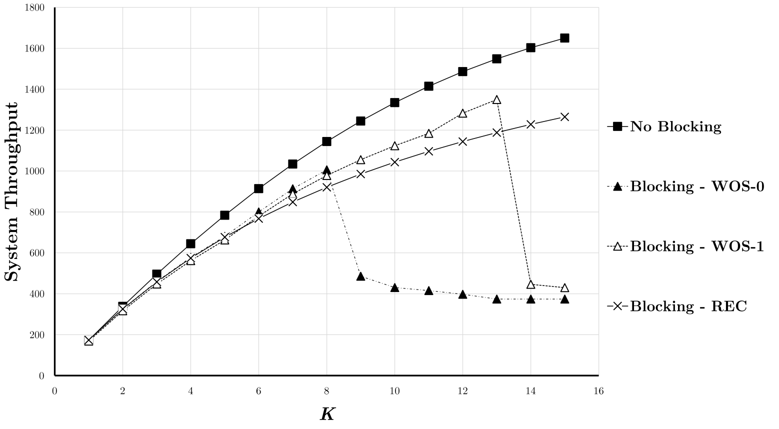


Figure 3.14: System throughput using different blocking policies

in a situation in which all four rack sections are full with a robot waiting on top of each. Therefore, the ninth robot will definitely be delayed, resulting in a sudden decrease in the system throughput. Using the WOS-1 policy pushes the point of sharp fall in throughput to a later stage. This phenomenon can be largely mitigated with the REC policy, where the robots circulate in the system while waiting for their rack section to become available. Consequently, they never block other moving robots.

If we compare the two policies, the REC has a lower system throughput, especially when the waiting time is less than the recirculation time and the number of robots is small. However, the REC policy can achieve a higher system throughput without worrying about the blocking delays when the number of robots in the system is increased. Therefore, depending on the number of robots in the system and the desired throughput level, the company can choose the most suitable policy.

3.7 Cost-Performance Comparison of the Vertical and Horizontal System

Building on the previous sections, we now have the tools to compare the performance of the vertical and the horizontal systems. Because the vertical system is primarily used for handling product totes, we also compare its performance with a horizontal system for handling product totes, see Marchet et al. (2012). A horizontal system consists of several aisles with multiple lifts each serving one or multiple aisles. The shuttles are either aisle-captive or pooled among aisles. They can also be either tier-captive or tier-to-tier. An L/U point is located at the bottom of each lift. The most comparable horizontal system to the vertical system is a system consisting of several aisles, where each aisle consists of two single-deep storage racks, with a dedicated lift and tier-to-tier shuttles. An L/U point is located at one end of each aisle (Figure 3.15). An extra L/U point can be added at the other end of the aisle, but then an extra lift needs to be installed as well. The lift can be either discrete or continuous. A discrete lift can only transport one shuttle at a time, while the continuous lift (i.e., a sort of vertical conveyor) can transport multiple shuttles simultaneously. However, the discrete lift is faster than a continuous lift. The reason is that a discrete lift only needs to transport one unit at a time, while a continuous lift needs to transport many units at a time which requires much more energy (Qimarox BV, 2013; Marchet et al., 2012). It should be noted that the shuttles can reach the first tier without needing a lift. However, a lift is needed to access the remaining tiers for the vertical movement. In this horizontal system, each aisle operates independently. Therefore, for a fair comparison, we analyze the performance of a single aisle of the vertical system with a single aisle of the horizontal system.

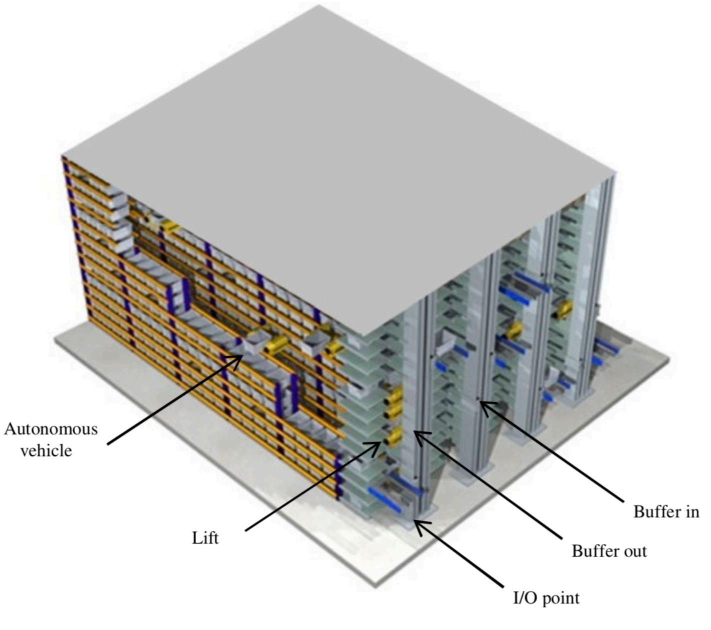


Figure 3.15: Horizontal system (Marchet et al., 2013)

To be in line with the vertical system analysis, we make similar assumptions for analyzing the horizontal system:

1. *Dual-command cycle orders for shuttles:* We assume all requests are executed in a dual-command cycle.

2. *Same tier:* We assume that the dual-command cycle orders are always for a specific tier. i.e., there is always a storage location available in the same tier from which we want to retrieve the new item.

3. *Uniform assignment:* We assume that storage and retrieval locations are assigned uniformly in the system.

4. *One shuttle per tier:* We assume only one shuttle can access each tier at a certain time in order to avoid the risk of having a deadlock in the system.

5. *Constant velocity:* We ignore acceleration and deceleration of shuttles and discrete lift during movements in our analysis, and shuttles and lift have a constant velocity.

6. *Dwell point:* We assume that the Point of Service Completion (POSC) dwell point policy for the discrete lift. We assume the shuttles dwell at the L/U point.

In our analysis, we compare the maximum system throughput of the two systems for a given K, assuming there is enough buffer space available in each tier of the horizontal system and each rack section of the vertical system. Then we calculate the operating cost associated with the system throughput for each system to investigate which system achieves a higher throughput at a lower operating cost. This cost-performance analysis can guide decision-makers to choose the right system based on performance needs and budget constraints. We estimate the performance of the horizontal system by developing a closed queuing network model with unlimited buffer space and estimate the performance of the vertical system by using the model developed in Section 3.4.1. The details of the model for the horizontal system are presented in Section 3.9.6.

The operating cost is evaluated on the basis of the annualized cost as a sum of three items: equipment cost, floor space cost, and picker cost.

$$Cost = (C_s + C_r)K + C_l N_{LU} + C_A A + C_p N_{LU} \tag{3.20}$$

where:

- C_s: annualized cost of a shuttle [€/year]
- C_l: annualized cost of a lift [€/year]
- C_r: annualized cost of a robot [€/year]
- C_p: annual cost of a picker [€/year]
- C_A: annual floor space cost [€/(m^2year)]
- A: floor space [m^2]
- K: number of robots or shuttles in the system

- N_{LU}: number of L/U points in the system

Note that C_s and C_l are zero for the vertical system, and C_r is zero for the horizontal system. The rack cost is assumed to be the same for both systems and therefore is ignored in our analysis. A is equal to $(w(depth))N_C$ in both the vertical and horizontal system. Table 3.11 shows the parameters of both systems. The shuttles and the discrete lift velocity parameters are taken from MWPVL International (2013). The continuous lift velocity is obtained by analyzing a video of an implementation by Qimarox BV (2013). Different costs are listed in Table 3.12 (courtesy of Dynamis B.V, Marchet et al. (2012) and experts' opinions).

Table 3.11: System parameters

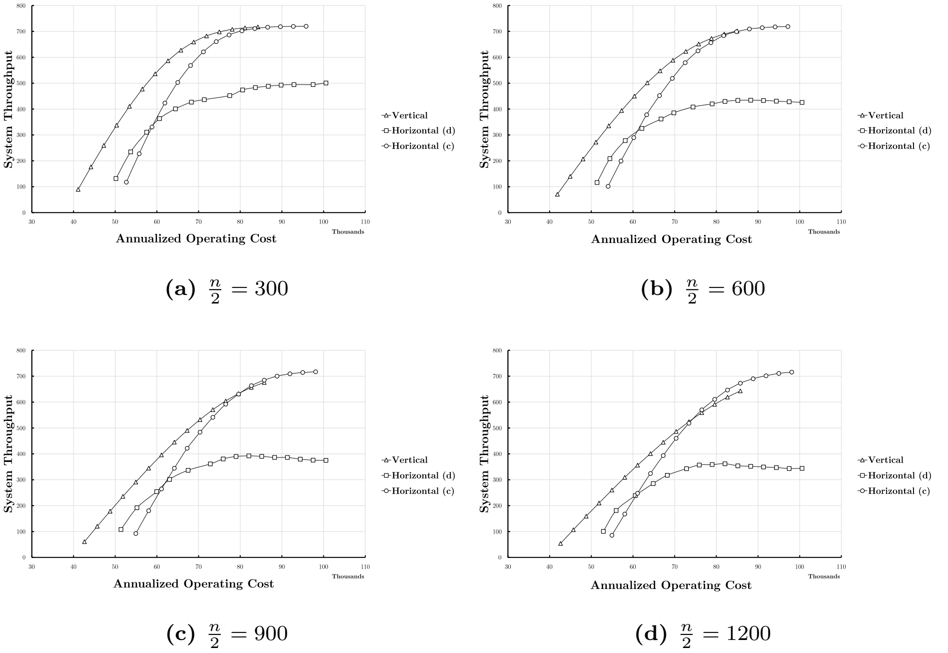
System	$n/2$	w	h	depth	v_r	v_s	v_l(disc)	v_l(cont)	τ_l, τ_u	τ_l, τ_u(lift)	τ_{LU}
Vertical System	300,600,900,1200	80 cm	32 cm	2.7 m	1 m/s	N/A	N/A	N/A	1.5 sec	N/A	5 sec
Horizontal System	300,600,900,1200	80 cm	32 cm	2.7 m	N/A	3.2 m/s	3.2 m/s	1 m/s	2 sec	3 sec	5 sec

Table 3.12: Unit costs

Cost item	Unit of measure	Value	Expected Life
Shuttle	€	15,000	7 years
Discrete Lift	€	50,000	7 years
Continuous Lift	€	60,000	7 years
Robot	€	15,000	7 years
Floor Space	€/(m^2.year)	70	N/A
Picker	€/year	35,000	N/A

We assume that there is one L/U point available in the systems. We first determine the layout that maximizes the throughput, with a given storage capacity ($n/2 = 300, 600, 900, 1200$), given number of robots ($K = 1, ..., 15$), and a system type. Then, the annualized cost of the configuration is calculated using the costs presented in Table 3.12, and assuming seven years of service and a 10% interest rate. Figure 3.16 illustrates the system throughput versus the annualized operating cost.

The vertical system always outperforms the horizontal system with discrete lift (horizontal-d), both in terms of operating costs and system throughput. No matter what the number of storage locations in the system is, the vertical system always has lower operating costs compared to the horizontal-d system for the same throughput. Furthermore, the system throughput in the horizontal-d system saturates as the lift becomes the bottleneck in the system. Therefore, the upper bound for the throughput capacity depends on the capacity of the lift. For instance, when $n/2 = 300$, the throughput capacity cannot be increased to more than 500 dual-command cycles per hour because the lift becomes the bottleneck at this point. Note that the capacity of the lift depends on the height of the rack, and because the rack is optimized for every individual instance, increasing the number of robots can lead to a different rack structure with different lift capacity. However, in the vertical system,



(a) $\frac{n}{2} = 300$

(b) $\frac{n}{2} = 600$

(c) $\frac{n}{2} = 900$

(d) $\frac{n}{2} = 1200$

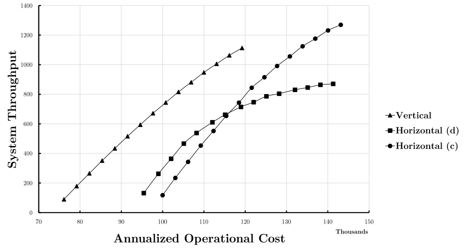
Figure 3.16: System throughput vs. annual operating cost - one L/U point

the system throughput can be increased until the picker becomes the bottleneck, at 720 dual-command cycles per hour.

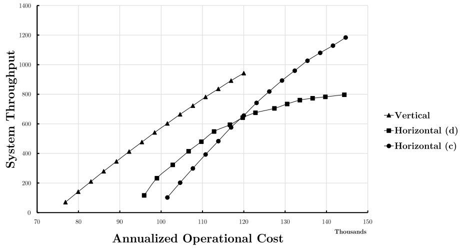
Compared to the horizontal system with continuous lift (horizontal-c), the horizontal-d system performs better when a lower system throughput is required. For a required system throughput of up to about 240 dual-command cycles per hour, the horizontal-d system has lower operating costs, while the horizontal-c system has a lower operating cost when a system throughput of more than about 390 dual-command cycles per hour is needed. Between a system throughput of 240 and 390, either system can perform with a lower operating cost depending on the number of storage locations in the system.

The performance of the vertical system also depends on the number of storage locations in the system. When $n/2 = 300$, the vertical system always produces a given system throughput at a lower operating cost compared to the horizontal-c system. However, as the number of storage locations increases, the operating cost of the vertical system increases more than the horizontal-c system. As a result, when $n/2 = 1200$ the vertical system has a lower operating cost when the required system throughput is at most 540 dual-command cycles per hour. Beyond this threshold, the horizontal-c system has a lower operating cost.

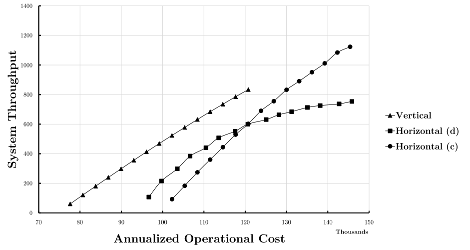
It is possible to carry out the same analysis assuming two L/U points are available in each system (at either end of the aisle). The results are presented in Figure 3.17.



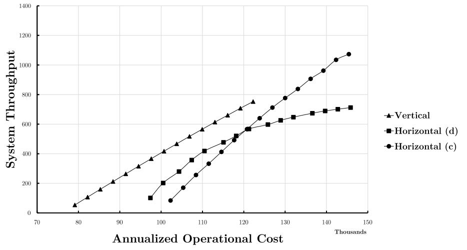
(a) $\frac{n}{2} = 300$



(b) $\frac{n}{2} = 600$



(c) $\frac{n}{2} = 900$



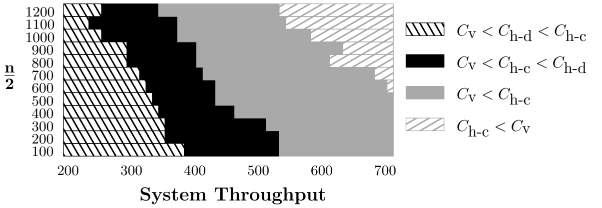
(d) $\frac{n}{2} = 1200$

Figure 3.17: System throughput vs. annual operating cost - two L/U points

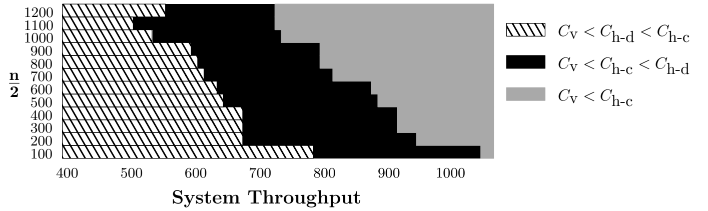
In this situation, the vertical system always has a lower operating cost compared to both horizontal systems. This is because an additional L/U point in the horizontal system requires the installation of an extra lift, which increases the operating cost significantly. For horizontal-c and horizontal-d systems, similar to the scenario with one L/U point, the operating cost depends on the number of storage locations in the system. When $n/2 = 300$, the horizontal-d system can generate a system throughput of up to 380 dual-command cycles per hour, with lower operating cost compared to the horizontal-c system. The number decreases to 560 when $n/2 = 1200$. Similar to the one L/U scenario, the system throughput of the horizontal-d system saturates sooner than the other systems at around 1060 dual-command cycle per hour because the lifts become the bottleneck.

Figure 3.18 summarizes the cost comparison of the three systems. In the figure, the annual operational costs of the vertical system (C_{V}), the horizontal-c system ($C_{\text{h-c}}$), and the horizontal-d system ($C_{\text{h-d}}$) are compared depending on the number of storage locations and the desired system throughput. Note that the throughput of the horizontal-d system cannot go beyond the black region in each scenario, because the lift becomes at the interface of the black and the gray regions.

Analysis and obtained insights: Except for one scenario, the vertical system produces similar throughput capacity with a lower annualized operating cost compared to the hori-



(a) One L/U point



(b) Two L/U points

Figure 3.18: Summary of the cost comparison of the three systems

zontal system. This cost difference can be explained by the fact that the horizontal system uses two material handling resource, lift as well as shuttles, whereas the vertical system only uses the rack climbing robots. The cost difference is even more notable when the systems have two L/U points. Because introducing an additional L/U point in the vertical system does not require the installation of any additional material handling system, its operating cost to achieve a certain throughput capacity is significantly lower even compared to the horizontal system with a continuous lift.

As mentioned earlier, discrete lifts are faster than continuous lift. However, the shuttles can immediately utilize the continuous lifts while they need to wait for the discrete lift. Therefore, when a lower throughput capacity level is required (which translates to a lower number of shuttles in the system), the shuttles waiting time for the discrete lift is lower than the shuttles travel time with continuous lift. Thus, the horizontal-d system has a better performance with a similar cost. On the other hand, as the desired throughput capacity level increases (and consequently the number of shuttles in the system increases), the shuttles waiting time for the discrete lift will be higher than the shuttle travel time with the continuous lift. Therefore, the horizontal-c system produces better performance at a similar cost.

3.8 Conclusion

In this paper, we study a vertical storage and retrieval system. This is a new robotic-based system which eliminates the need for lifts for the vertical transportation. Robots in this system have the ability to move horizontally and vertically independently and access all the storage locations within the aisle. We develop analytical models to analyze the performance of the vertical system.

First, we assume there are infinite buffer locations available in each rack section and therefore no blocking delays occur in the system. We develop a closed queuing network and estimate the system throughput by using AMVA. We then develop a system dimension optimization model to obtain the optimal system layout for the maximum system throughput. The results show that when using the provided system parameters, the optimal ratio of height-to-width (i.e., number of tiers divided by the number of rack sections) is around one. Next, we relax the assumption of having unlimited buffer locations in each rack section to estimate the effect of blocking delays on the system performance. We propose a REC waiting policy for the blocked robots, as opposed to the more obvious WOS policy in which the robots wait on top of the occupied rack section until it is unoccupied. We develop two closed queuing networks corresponding to each policy to estimate the performance of the system. We use an approximate method proposed by Akyildiz (1988) to estimate the system throughput in the WOS network. In the case of REC network, we develop a new approximation technique by using the jump-over blocking protocol and an iterative algorithm based on MVA to estimate the performance of the system. This results in a very accurate estimation of the system throughput of the real system. Comparing the results of the two policies, the WOS policy has a slight advantage when the number of robots in the system is small. However, increasing the number of robots results in a sharp decrease in the system throughput which can be mitigated by adopting the REC policy.

We then compare the performance of the vertical system with the horizontal system. The horizontal system is modeled by a closed queuing network with an unlimited buffer and the system throughput is estimated using AMVA. We compare the cost-performance of the vertical system, the horizontal system with a discrete lift, and the horizontal system with a continuous lift. The results indicate that when there is one L/U point in the system, the vertical system outperforms the horizontal-d system in both operating costs and system throughput. However, compared to the horizontal-c system, its performance depends on the number of storage locations in the system.

When there are two L/U points in the system, the vertical system always has a lower operating cost compared to both the horizontal systems. Horizontal systems require an extra lift when there are two L/U points, which increases operating costs significantly.

3.9 Appendix

3.9.1 Approximate Mean Value Analysis (AMVA)

Approximate Mean Value Analysis or AMVA is an approximation technique which extends the regular MVA (Reiser & Lavenberg, 1980) to include nodes with a generally distributed service time. The MVA algorithm exploits two simple laws: a) *Little's Law* which expresses the relation between mean residence time, throughput and mean number of jobs in a node of the overall network, and b) *Arrival theorem* which says that upon arrival at a node i, a job observes the system in the steady state with one job less (Bolch et al., 2006). Here we present the AMVA method for a single class closed queuing network based the algorithm presented in Zijm (2002). The notation used in the algorithm is presented in Table 3.13.

Table 3.13: Notations AMVA

Notation	*Description*
k	number of jobs in the network
K	total number of jobs in the system
i	nodes index: 1,2,...,N
$T_i(k)$	mean residence time at node i when there are k jobs in the network
$Q_i(k)$	mean number of jobs in the queue at node i when there are k jobs in the network
$U_i(k)$	the probability that all servers are busy at node i when there are k jobs in the network
m_i	number of servers in node i
$X(k)$	system throughput when there are k jobs in the network
$\pi_i(q\|k)$	marginal probability of having q jobs in node i when there are k jobs in the system
V_i	visit ratio for node i
μ_i^{-1}	expected service time of node i
$\mu_{i,rem}^{-1}$	expected remaining service time of node i

The residence time at node i has three components: 1) the time until the first departure, 2) the time to clear all the jobs in the queue, and 3) the service time of the job itself. The first component is calculated by the product of the probability that all servers are busy when the job arrives and the remaining service time. Note that since the service time is not exponentially distributed anymore, the average remaining service time is not equal to the average service time. It is shown that the average remaining service time given that all servers are busy is calculated by the following equation (Zijm, 2002):

$$\mu_{i,rem}^{-1} = \frac{1}{\mu_i.m_i}\left(\frac{m_i + cv_i^2}{m_i + 1}\right) \tag{3.21}$$

The second component is calculated by the product of the number of jobs in the queue when the job arrives, and the average service time divided by the number of servers. And the last component is the expected service time for the arriving job. Therefore, the mean residence time at node i when there are k jobs in the network is calculated as follows:

$$T_i(k) = \mu_{i,rem}^{-1}.U_i(k-1) + \frac{\mu_i^{-1}}{m_i}.Q_i(k-1) + \mu_i^{-1} \tag{3.22}$$

In which:

$$U_i(k-1) = \sum_{j=m_i}^{k-1} \pi_i(j|k-1) \tag{3.23}$$

and

$$Q_i(k-1) = \sum_{j=m_i+1}^{k-1} (j-m_i)\pi_i(j|k-1) \tag{3.24}$$

In case of infinite server node, $U_i(k-1)$ and $Q_i(k-1)$ is equal to zero and therefore, $T_i(k) = \mu_i^{-1}$.

Now we can formulate the algorithm as follow:

Step 1: Initialization. For all $i = 1, ..., N$, $\pi_i(0|0) = 1$.

Step 2: Start iteration over the number of jobs $k = 1, ..., K$

Step 3: Calculate the mean response time of a job at node i, $T_i(k)$, using Equation 3.22.

Step 4: Compute the system throughput and throughput of each node using Equations 3.25 and 3.26 respectively:

$$X(k) = \frac{k}{\sum_{i=1}^{N} V_i E[T_i(k)]} \tag{3.25}$$

$$X_i(k) = V_i X(k) \tag{3.26}$$

Step 5: Calculate the conditional probabilities by Equations 3.27 and 3.28

$$\pi_i(0|k) = 1 - \sum_{j=1}^{k} \frac{X_i(k)}{\mu_i \alpha_i(j)} \pi_i(j-1|k-1) \tag{3.27}$$

$$\pi_i(j|k) = \frac{X_i(k)}{\mu_i \alpha_i(j)} \pi_i(j-1|k-1) \tag{3.28}$$

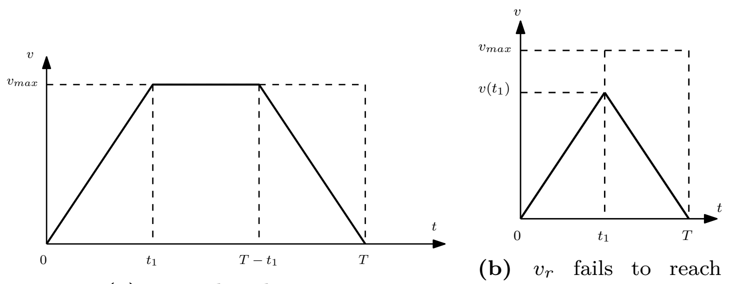
In which $\alpha_i(j)$ is:

$$\alpha_i(j) = \begin{cases} j & \text{if } j \le m_i \\ m_i & o.w \end{cases} \tag{3.29}$$

Step 6: Return to *Step 3* until $k = K$.

3.9.2 Average Velocity Calculation

Because the robot stops twice to load and unload tote in the rack section, its average velocity is much lower than its maximum velocity including acceleration and declaration. We use a procedure similar to Lerher et al. (2010) to calculate the average velocity of the robot. Depending on the storage and retrieval location, the robot's velocity either reaches its peak or not. Therefore, the velocity-time relationship of the robot is one of the graphs in Figure 3.19.



(a) v_r reaches the v_{max}

(b) v_r fails to reach v_{max}

Figure 3.19: Velocity-time relationship of the robot

We now present the average velocity for the robot in each scenario.

Velocity $v(t_1)$ is equal to v_{max}:

In this scenario, the velocity expression in dependence of time is presented in the following equation:

$$v(t) = \begin{cases} at & 0 < t < t_1 \\ v_{max} & t_1 < t < T - t_1 \\ -a(t-T) & T - t_1 < t < T \end{cases} \tag{3.30}$$

In which a is the acceleration. Now we can calculate the distance in depend of time by taking an integral from the velocity:

$$\begin{aligned} d(t) = \int_0^t v(t)dt &= \int_0^{t_1} atdt + \int_{t_1}^{T-t_1} v_{max}dt + \int_{T-t_1}^{T} -a(t-T)dt \\ &= \frac{at_1^2}{2} + v_{max}(T - 2t_1) + \frac{at_1^2}{2} \end{aligned} \tag{3.31}$$

It is clear from Figure 3.19a that $t_1 = v_{max}/a$. By substituting it in Equation 3.31 we have:

$$d(T) = v_{max}.T - \frac{V_{max}^2}{a} \tag{3.32}$$

Therefore the average velocity is:

$$v_{avg} = \frac{d(T)}{T} = v_{max} - \frac{v_{max}^2}{T.a} \tag{3.33}$$

Velocity $v(t_1)$ is less than v_{max}:

In this scenario, the velocity expression in dependence of time is presented in the following equation:

$$v(t) = \begin{cases} at & 0 < t < t_1 \\ -a(t-T) & t_1 < t < T \end{cases} \tag{3.34}$$

Now we can calculate the distance in depend of time by taking an integral from the velocity:

$$\begin{aligned} d(t) = \int_0^t v(t)dt &= \int_0^{t_1} atdt + \int_{T-t_1}^{T} -a(t-T)dt \\ &= \frac{at_1^2}{2} + \frac{a(T-t_1)^2}{2} \end{aligned} \tag{3.35}$$

It is clear from Figure 3.19a that $t_1 = T/2$. By substituting it in Equation 3.35 we have:

$$d(T) = \frac{aT^2}{4} \tag{3.36}$$

Therefore the average velocity is:

$$v_{avg} = \frac{d(T)}{T} = \frac{aT}{4} \tag{3.37}$$

3.9.3 MVA for Jump-Over Network

Step 1: For all $i = 1, 2, ..., N_C$, initialize the blocking probabilities: $p_{b_i}^{current} = arbitrary$ *number between* 0. *and* 1.

Step 2: For all $j = LU, F, UP_i, RS_i, D_i, B_i$ and $i = 1, 2, ..., N_C$, initialize $L_j(0) = 0$, $\pi_{RS_i}(0|0) = 1$, $\pi_{RS_i}(1|0) = 0$.

Step 3: Start iteration over the number of jobs $k = 1, ..., K$.

Table 3.14: Notations for jump-over network

Notation	*Description*
k	number of robots in the network
K	total number of robots in the system
j, m, n	nodes in the network: LU, F, UP_i, RS_i, D_i, B_i
i	rack section index: 1, 2, ..., N_C
$T_j(k)$	mean residence time of node j when there are k jobs in the network
$L_j(k)$	mean number of robots at node j when there are k jobs in the network
$X(k)$	overall throughput of the system when there are k jobs in the network
$P_{m,n}$	routing probability of going from node m to node n
p_{b_i}	blocking probability of node RS_i
$U_j(k)$	utilization of node j when there are k jobs in the network
$\pi_j(q\|k)$	marginal probability of having q jobs in node j when there are k jobs in the system
V_j	visit ratio for node j
μ_j^{-1}	expected service time of node j

Step 4: Calculate mean throughput time for each node:

$$T_j(k) = \begin{cases} \frac{1}{\mu_j}(1 + L_j(k-1)) & \text{if } j = LU \\ \frac{1}{\mu_j} & \text{if } j = F, Up_i, D_i, B_i \\ \pi_j(1|k-1).0 + (1 - \pi_j(1|k-1))\, \frac{1}{\mu_j} & \text{if } j = RS_i \end{cases}$$

Step 5: Calculate the system throughput:

$$X(k) = \frac{k}{\sum\limits_j V_j T_j(k)}$$

Step 6: Calculate mean queue length for each node:

$$L_j(k) = V_j X(k) T_j(k)$$

Step 7: Update marginal probabilities that the node RS_i is occupied:

$$\text{(I) } \pi_{RS_i}(1|k) = \frac{V_{RS_i} X(k)}{\mu_{RS_i}} \pi_{RS_i}(0|k-1)$$

$$\text{(II) } \pi_{RS_i}(0|k) = 1 - \pi_{RS_i}(1|k)$$

Step 8: Repeat *Steps 4 to 6*, until $k = K$

Step 9: Calculate new blocking probabilities for node RS_i:

$$p_{b_i}^{new} = \pi_{RS_i}(1|K)$$

Step 10: If $|p_{b_i}^{new} - p_{b_i}^{current}| > \epsilon$, update $p_{b_i}^{current} = p_{b_i}^{new}$ and return to *Step 2*, otherwise, end.

3.9.4 Working Example for Estimating the Performance of the System with REC Block Prevention Policy

In this part we will illustrate a working example to estimate the performance of the system with a REC blocking prevention policy. We estimate the performance of the system consists of two rack section and four tiers. assume to have a system with two rack section and 4 tiers. Table 3.15 presents the rest of the parameters.

Table 3.15: System parameters

$n/2$	N_T	N_C	w	h	K	v_r	τ_l, τ_u	τ_{LU}
8	4	2	80 cm	32 cm	5	1 m/s	1.5 sec	3.6 sec

Figure 3.20 presents the corresponding closed queuing network of the system.

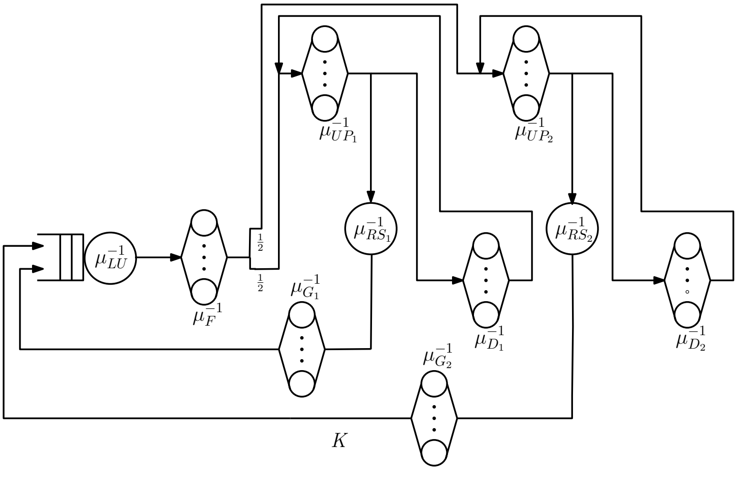


Figure 3.20: Corresponding closed queuing network with the REC block prevention policy.

The corresponding jump-over approximation of the closed queuing network is presented in Figure 3.21.

Using the parameters on Table 3.15 and Equations 3.5, 3.7, 3.8, 3.9, and 3.10 we estimated the service times for all the nodes of the network. Furthermore, we obtain the expected visit ratios of the nodes in the network by the described method in Section 3.5.3.

We now use the MVA algorithm for the jump-over network. We initialize $p_{b_1} = p_{b_2} = 0$ and $\epsilon = 0.0001$. The results are presented in Table 3.17.

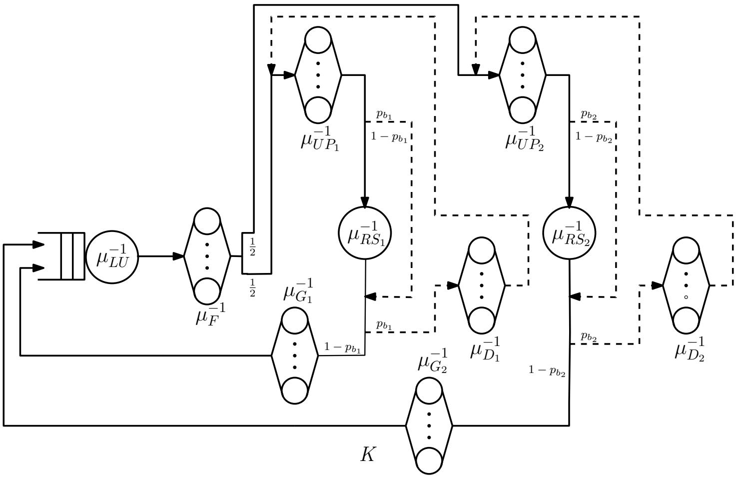


Figure 3.21: Jump-over approximation of the closed queuing network

Table 3.16: Modeling parameters

Node	μ_{LU}^{-1}	μ_F^{-1}	$\mu_{UP_1}^{-1}$	$\mu_{UP_2}^{-1}$	$\mu_{G_1}^{-1}$	$\mu_{G_2}^{-1}$	$\mu_{D_1}^{-1}$	$\mu_{D_2}^{-1}$	$\mu_{RS_1}^{-1}$	$\mu_{RS_2}^{-1}$
Time (sec)	3.6	1.6	2.08	2.88	2.72	3.52	5.92	5.12	5.31	5.29
Visit Ratio	1	1	$\frac{1}{2(1-p_{b_1})}$	$\frac{1}{2(1-p_{b_2})}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{p_{b_2}}{2(1-p_{b_1})}$	$\frac{p_{b_2}}{2(1-p_{b_2})}$	$\frac{1}{2(1-p_{b_1})}$	$\frac{1}{2(1-p_{b_2})}$

Table 3.17: Estimated performance results

Throughput (X)	Blocking Probability- RS_1	Blocking Probability -RS_2
684.41	0.4651	0.4658

3.9.5 Deriving Optimal Layout of the Vertical System by Using Travel Time Expressions

Taking the expected value from Equation 3.1 and doing some simplifications results in Equation 3.39:

$$\begin{aligned} E[CT_{dc}] =& \tau_{LU} + \frac{2w}{v_r} + \frac{N_T h}{v_r} + \frac{(N_C+1)w}{2v_r} \\ &+ \frac{(N_T+1)h}{2v_r} + \tau_u + \frac{(N_T+1)h}{3v_r} + \tau_l + \frac{(N_T+1)h}{2v_r} \\ &+ \frac{(N_C+1)w}{2v_r} + \frac{w}{v_r} + \frac{h}{v_r} + \frac{w}{v_r} \end{aligned} \tag{3.38}$$

$$E[CT_{dc}] = \tau + \frac{w}{v_r}\left(2 + \frac{N_C+1}{2} + \frac{N_C+1}{2} + 1 + 1\right) + \frac{h}{v_r}\left(N_T + \frac{N_T+1}{2} + \frac{N_T+1}{3} + \frac{N_T+1}{2} + 1\right) \tag{3.39}$$

in which

$$\tau = \tau_{LU} + \tau_u + \tau_l$$

The total storage number is $n = 2N_T N_C$. Therefore $N_C = \frac{n}{2N_T}$. By substituting N_C in Equation 3.39 we have,

$$E[CT_{dc}] = y(N_T) = \tau + \frac{w}{v_r}\left(5 + \frac{n}{2N_T}\right) + \frac{h}{v_r}\left(\frac{7}{3}N_T + \frac{7}{3}\right) \tag{3.40}$$

Taking the derivative of Equation 3.40 with respect to N_T and equaling it to zero results in:

$$N_T = \sqrt{\frac{3nw}{14h}} \tag{3.41}$$

$$N_C = \frac{n}{2N_T} \tag{3.42}$$

Therefore:

$$\frac{N_T}{N_C} = \frac{3w}{7h} \tag{3.43}$$

Which in our case is, $\frac{N_T}{N_C} = \frac{3.80}{7.32} = 1.07$

3.9.6 Horizontal System with Unlimited Buffer Locations inside each Tier

Figure 3.22 presents the side view of a single aisle of the horizontal system is presented.

Figure 3.23a illustrate how an item is stored in the horizontal system with a discrete lift. First, the shuttle waits for the lift to pick it up (1). Next, the shuttle is transported vertically with the lift in front of the desired tier (2). Then, the shuttles move horizontally in the tier and stores the item (3). For retrieval, which is illustrated in Figure 3.23b, the shuttle moves to the retrieval location in the tier (1), picks the item and transports it to the front of the tier (2), and waits for the lift (3). The lift brings the shuttle down towards the L/U point for offloading the item (4). The movement of the shuttle is identical in the horizontal system with continuous lift, except that the shuttles do not wait for the lift and can immediately use it to go up or down.

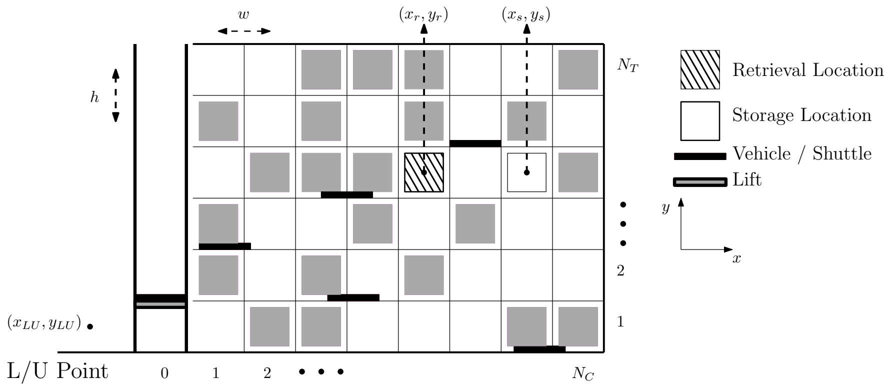
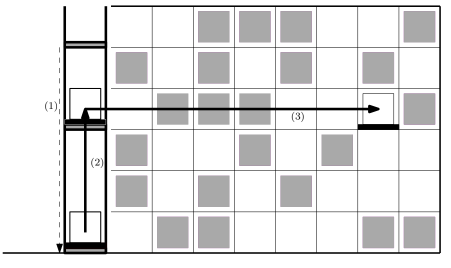
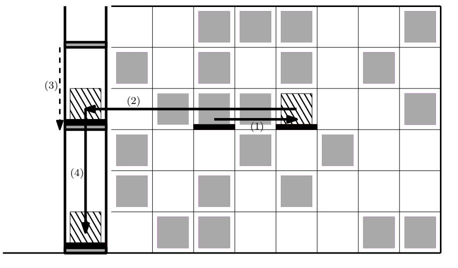


Figure 3.22: Side view of a single aisle of the horizontal system



(a) Storage Process



(b) Retrieval Process

Figure 3.23: Storage and retrieval process in the horizontal system

Horizontal systems have been extensively studied in the literature (see Section 3.2). Here we present the closed queuing model we developed to estimate the performance of this system.

The horizontal system can use a discrete or a continuous lift for the vertical movement, each resulting in a slightly different model. Because the discrete lift can only serve one shuttle at a time, it is modeled as a single server queue with generally distributed service time. In contrast, the continuous lift can serve multiple shuttles simultaneously; therefore, it is modeled as an infinite server queue.

Horizontal System with Discrete Lift

The L/U point is modeled as a single server queue with an exponentially distributed service time. Upon service completion at L/U point, the shuttle either directly goes to the first tier or takes the lift to go to other tiers. Because tiers are uniformly selected, with probability $\frac{1}{N_T}$ the shuttle goes from the L/U point to the first tier, and with probability $\frac{N_T-1}{N_T}$, it takes the lift to go to other tiers. The load retrieval and storage process within each tier is modeled as a single server queue with generally distributed service time. Upon service completion in the tier, the shuttle is directed to the L/U point. If the shuttle is in the first tier, it does not require the lift to return to the L/U point. But if the shuttle is in any other tiers, it needs to use the lift to descend to the L/U point. The corresponding closed queuing network is illustrated in Figure 3.24.

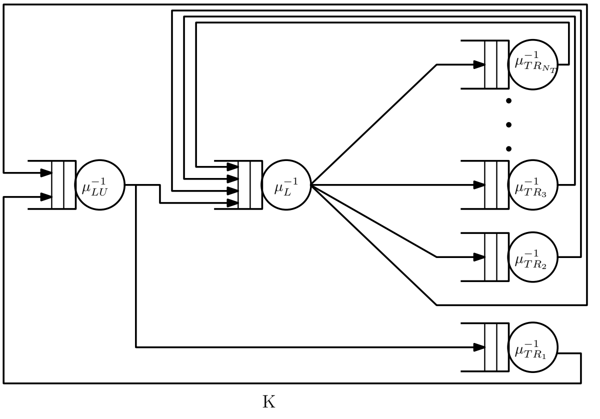


Figure 3.24: losed queuing network - the horizontal system with discrete lift

Service Times Expressions

The term μ_{LU}^{-1} depends on the speed of the picker and $\mu_{TR_i}^{-1}$ is based on the travel time expression for a shuttle to perform a dual-command in tier, $i = 1, 2, ..., N_T$. The storage and retrieval locations in each tier follow a uniform distribution from N_C available storage columns in each tier. Using Figure 3.22, the time expression to perform a dual-command cycle in tier i is presented in Equation 3.44.

$$T_{TR_i} = \frac{|x_0 - x_s|}{v_s} + \tau_u + \frac{|x_s - x_r|}{v_s} + \tau_l + \frac{|x_r - x_0|}{v_s} \tag{3.44}$$

Therefore, the expected service time and the squared coefficient of variation for the node TR_i is:

$$\mu_{TR_i}^{-1} = E[T_{TR_i}] \tag{3.45}$$

$$cv_{TR_i}^2 = \frac{Var[T_{TR_i}]}{[E[T_{TR_i}]]^2} \tag{3.46}$$

The time expression for the lift consists of two parts. We assume with probability 1/2 the lift is in a retrieval process and with probability 1/2 is in a storage process, i.e., half of the time the lift goes from its dwell position (y_d) to the L/U point to pick up a shuttle and to bring it to the desired tier (Equation 3.47), and half of the time the lift goes from its dwell position to another tier to pick up the shuttle and to bring it down to the L/U point (Equation 3.48).

$$T_{L_S} = Y_{LS} = \frac{|y_d - y_{LU}|}{v_l} + \tau_l + \frac{|y_{LU} - y_s|}{v_l} + \tau_u \tag{3.47}$$

$$T_{L_R} = Y_{LR} = \frac{|y_d - y_r|}{v_l} + \tau_l + \frac{|y_r - y_{LU}|}{v_l} + \tau_u \tag{3.48}$$

As we mentioned earlier, y_s and y_r are random variables with a uniform distribution between two and N_T. Also, the lift uses the POSC dwell policy, i.e., the lift dwells either at the L/U point in the first tier, after completing the retrieval process, or dwells in front of a tier, after elevating a shuttle to perform the storage process. Hence, y_d is a random variable, which with probability of 1/2 is at tier one and with probability of $(1/2)(1/(N_T - 1))$ is at any other tier. So the service time of the lift has four scenarios as illustrated in Figure 3.25.

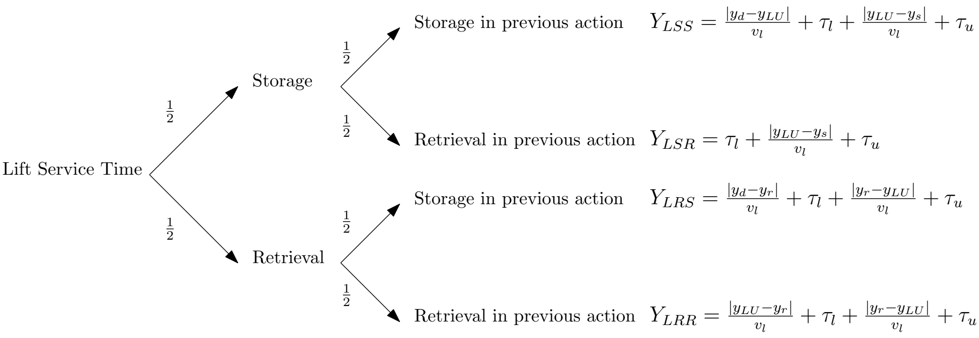


Figure 3.25: Lift service time

Y_{LSS}, Y_{LSR}, Y_{LRS}, and Y_{LRR} are random variables with probability density function $f_{LSS}(y)$, $f_{LSR}(y)$, $f_{LRS}(y)$, and $f_{LRR}(y)$ respectively. The total service time of the lift is a random variable Y_L with probability density function $g_L(y)$, which is equal to $f_{LSS}(y)$ with probability 1/4, $f_{LSR}(y)$ with probability 1/4, $f_{LRS}(y)$ with probability 1/4, and $f_{LRR}(y)$ with probability 1/4. The expected value of Y_L is calculated in Equation 3.49.

$$\begin{aligned} E[Y_L] &= \int y g_L(y) dy \\ &= \frac{1}{4}\int y f_{LSS}(y)dy + \frac{1}{4}\int y f_{LSR}(y)dy + \frac{1}{4}\int y f_{LRS}(y)dy + \frac{1}{4}\int y f_{LSR}(y)dy \\ &= \frac{1}{4}E[Y_{LSS}] + \frac{1}{4}E[Y_{LSR}] + \frac{1}{4}E[Y_{LRS}] + \frac{1}{4}E[Y_{LRR}] \end{aligned} \tag{3.49}$$

For calculating the variance of Y_L, first we calculate the expected value of Y_L^2.

$$
\begin{aligned}
E[Y_L^2] &= \int y^2 g_L(y)dy \\
&= \frac{1}{4}\int y^2 f_{LSS}(y)dy + \frac{1}{4}\int y^2 f_{LSR}(y)dy + \frac{1}{4}\int y^2 f_{LRS}(y)dy + \frac{1}{4}\int y^2 f_{LSR}(y)dy \\
&= \frac{1}{4}\left(Var[Y_{LSS}] + [E[Y_{LSS}]]^2\right) + \frac{1}{4}\left(Var[Y_{LSR}] + [E[Y_{LSR}]]^2\right) \\
&+ \frac{1}{4}\left(Var[Y_{LRS}] + [E[Y_{LRS}]]^2\right) + \frac{1}{4}\left(Var[Y_{LRR}] + [E[Y_{LRR}]]^2\right) \qquad (3.50)
\end{aligned}
$$

Now the variance is

$$Var[Y_L] = E[Y_L^2] - [E[Y_L]]^2 \qquad (3.51)$$

Therefore the expected service time and the squared coefficient of variation for the node L is:

$$\mu_L^{-1} = E[Y_L] \qquad (3.52)$$

$$cv_L^2 = \frac{Var[Y_L]}{[E[Y_L]]^2} \qquad (3.53)$$

Horizontal System with Continuous Lift

As we described earlier, a continuous lift can provide service to more than one shuttle. Therefore, the shuttle does not need to wait and can immediately utilize the lift for the vertical movement. Consequently, we can separate the storage and retrieval action of the lift, and model each as an infinite server queue. The rest of the network is similar to the discrete lift network. The resulting closed queuing network is presented in Figure 3.26.

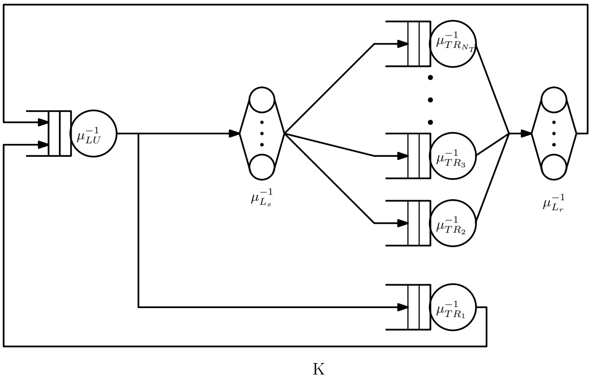


Figure 3.26: Closed queuing network - the vertical system with continuous lift

Service Times Expressions

The terms, μ_{LU}^{-1} and $\mu_{T_i}^{-1}$ take the same value as the one shown in the network for the system with discrete lift. The service time for the lift only depends on the original tier and the destination tier, therefore,

$$\mu_{L_s}^{-1} = \mu_{L_r}^{-1} = \tau_l + \frac{1}{N_T - 1} \sum_{i=2}^{N_T} \left[\frac{1}{N_T - 1} \left(\frac{i-1}{v_l} \right) h \right] + \tau_u \tag{3.54}$$

Solution Approach

Because L and TR_i nodes in the closed queuing network have generally distributed service times, we use AMVA to estimate the performance of the system.

3.9.7 Tabular Numerical Results

In this part we present the numerical data behind the figures.

Table 3.18: Data for the Figure 3.13

$n/2$	N_T/N_C	Throughput (DC/h) 5 Robots	Throughput (DC/h) 10 Robots
300	0.03	186.74	364.09
300	0.05	232.21	443.70
300	0.08	270.24	504.32
300	0.12	301.69	549.17
300	0.33	378.63	634.74
300	0.48	397.43	649.97
300	0.75	410.54	659.07
300	1.33	410.36	656.83
300	2.08	395.99	642.07
300	3.00	376.98	620.02
300	8.33	291.17	482.44
300	12.00	252.51	407.92
300	18.75	210.60	315.23
300	33.33	157.56	208.17
600	0.06	182.52	356.36
600	0.11	223.72	429.36
600	0.17	256.25	482.64
600	0.24	281.39	520.58
600	0.38	307.68	556.54
600	0.67	329.39	583.53
600	0.96	334.94	588.35
600	1.04	334.80	588.05
600	1.50	330.72	581.06
600	2.67	308.91	548.49
600	4.17	280.93	504.02
600	6.00	255.96	455.33
600	9.38	218.75	383.96
600	16.67	172.19	291.29
900	0.09	178.51	348.97
900	0.11	192.10	373.77
900	0.16	215.81	415.81
900	0.25	243.60	462.36
900	0.36	263.38	493.79
900	0.44	272.90	507.76
900	0.69	287.21	527.44
900	1.00	291.00	532.21
900	1.44	288.07	526.75
900	2.25	274.88	503.83
900	2.78	265.50	487.95
900	4.00	245.68	452.16
900	6.25	215.41	395.89
900	9.00	191.81	346.31
1200	0.12	174.66	341.89
1200	0.19	201.05	389.78

Continued on next page

Table 3.18 – *Continued from previous page*

$n/2$	N_T/N_C	Throughput (DC/h) 5 Robots	Throughput (DC/h) 10 Robots
1200	0.21	208.54	402.74
1200	0.33	232.05	443.15
1200	0.48	247.50	468.30
1200	0.52	250.26	472.68
1200	0.75	259.42	485.90
1200	1.33	260.46	486.31
1200	1.92	252.59	471.17
1200	2.08	249.68	467.26
1200	3.00	234.85	439.66
1200	4.69	210.85	393.37
1200	5.33	202.30	378.10
1200	8.33	175.37	323.21

Table 3.19: Data for the Figure 3.14

K	Throughput (DC/h) No Blocking	Throughput (DC/h) Blocking - WOS-0	Throughput (DC/h) Blocking - WOS-1	Throughput (DC/h) Blocking - REC
1	173.08	173.90	168.77	173.96
2	338.53	326.40	317.10	325.62
3	495.80	460.18	447.52	458.20
4	644.39	577.56	562.19	574.52
5	783.85	680.54	663.01	677.02
6	913.79	799.94	779.29	767.76
7	1033.90	913.24	888.62	848.47
8	1144.00	1006.05	978.12	920.58
9	1244.07	485.59	1055.43	985.29
10	1334.20	430.28	1123.63	1043.62
11	1414.66	415.84	1184.34	1096.39
12	1485.86	397.76	1282.81	1144.32
13	1548.37	374.17	1349.79	1188.00
14	1602.83	374.44	446.36	1227.96
15	1649.98	374.22	430.03	1264.61

Table 3.20: Data for the Figure 3.16 and Figure 3.18a

$n/2$	Vertical		Horizontal (d)		Horizontal (c)	
	Throughput	Cost	Throughput	Cost	Throughput	Cost
100	126.70	39593.08	151.86	49863.36	142.12	51917.41
100	244.70	42674.16	270.46	52944.44	271.10	54998.49
100	352.74	45755.25	354.90	56025.52	386.45	58079.58

Continued on next page

Table 3.20 – *Continued from previous page*

$n/2$	Vertical		Horizontal (d)		Horizontal (c)	
	Throughput	Cost	Throughput	Cost	Throughput	Cost
100	447.41	48836.33	412.02	60618.60	483.53	61160.66
100	528.09	51917.41	457.19	63699.69	563.47	64241.74
100	592.39	54998.49	489.03	66780.77	621.68	67322.82
100	641.14	58079.58	509.01	69861.85	663.28	70403.91
100	674.52	61160.66	520.36	72942.93	689.81	73484.99
100	695.75	64241.74	525.41	76024.02	705.12	76566.07
100	708.19	67322.82	527.57	79105.10	713.14	79647.15
100	714.62	70403.91	528.00	82186.18	717.04	82728.24
100	717.70	73484.99	525.63	85267.26	718.81	85809.32
100	719.09	76566.07	524.47	88348.35	719.57	88890.40
100	719.66	79647.15	522.77	91429.43	719.85	91971.48
100	719.88	82728.24	520.20	94510.51	719.95	94296.57
200	98.91	41105.08	140.22	49863.36	124.20	51917.41
200	193.73	44186.16	246.81	52944.44	240.35	54998.49
200	283.92	47267.25	323.80	57537.52	346.94	58079.58
200	367.98	50348.33	382.84	60618.60	442.70	61160.66
200	444.81	53429.41	423.02	64455.69	523.54	64241.74
200	513.03	56510.49	447.27	67536.77	588.56	67322.82
200	570.85	59591.58	460.92	70617.85	639.47	70403.91
200	618.34	62672.66	479.86	75966.93	672.83	73484.99
200	654.64	65753.74	494.33	79048.02	694.66	76566.07
200	680.42	68834.82	506.28	82129.10	707.43	79647.15
200	697.72	71915.91	513.38	85210.18	714.45	82728.24
200	708.33	74996.99	517.86	88291.26	717.76	85809.32
200	714.24	78078.07	519.97	91372.35	719.08	88890.40
200	717.36	81159.15	521.36	94453.43	719.66	91971.48
200	718.87	84240.24	521.63	97534.51	719.88	95052.57
300	89.62	41105.08	131.76	50165.76	117.64	52673.41
300	176.34	44186.16	234.67	53700.44	227.46	55754.49
300	259.34	47267.25	310.66	57537.52	330.83	58835.58
300	337.61	50348.33	363.41	60618.60	423.21	61916.66
300	410.62	53429.41	400.85	64455.69	502.91	64997.74
300	477.06	56510.49	426.94	68292.77	568.44	68078.82
300	536.01	59591.58	436.45	71373.85	621.00	71159.91
300	586.39	62672.66	452.31	77478.93	660.67	74240.99
300	627.45	65753.74	473.97	80560.02	686.41	77322.07
300	659.03	68834.82	483.06	83641.10	702.27	80403.15
300	682.40	71915.91	488.22	86722.18	710.93	83484.24
300	697.82	74996.99	492.73	89803.26	716.12	86565.32
300	708.00	78078.07	494.54	92884.35	718.35	89646.40
300	713.75	81159.15	494.77	97477.43	719.33	92727.48
300	717.03	84240.24	500.72	100558.51	719.75	95808.57
400	82.08	41105.08	126.13	50770.56	110.41	53429.41
400	161.77	44186.16	224.47	53851.64	216.92	56510.49
400	238.52	47267.25	296.30	58293.52	317.11	59591.58
400	311.15	50348.33	348.38	61374.60	403.54	62672.66
400	380.33	53429.41	381.61	66723.69	483.80	65753.74
400	444.38	56510.49	411.64	69804.77	549.23	68834.82
400	502.56	59591.58	427.96	72885.85	607.05	71915.91
400	554.38	62672.66	446.77	77478.93	645.92	74996.99

Continued on next page

Table 3.20 – *Continued from previous page*

$n/2$	Vertical		Horizontal (d)		Horizontal (c)	
	Throughput	Cost	Throughput	Cost	Throughput	Cost
400	598.41	65753.74	455.42	80560.02	676.07	78078.07
400	633.97	68834.82	458.64	83641.10	694.54	81159.15
400	662.54	71915.91	461.84	86722.18	707.51	84240.24
400	683.70	74996.99	460.32	89803.26	714.04	87321.32
400	698.00	78078.07	457.84	92884.35	717.28	90402.40
400	707.42	81159.15	454.65	95965.43	718.67	93483.48
400	713.44	84240.24	451.51	99046.51	719.52	96564.57
500	75.15	41861.08	120.28	51375.36	106.02	53429.41
500	148.53	44942.16	217.81	54456.44	208.85	56510.49
500	219.52	48023.25	285.18	58293.52	298.82	59591.58
500	288.07	51104.33	333.41	61374.60	386.20	62672.66
500	353.71	54185.41	358.75	68235.69	467.60	65753.74
500	415.01	57266.49	393.78	71316.77	533.57	68834.82
500	472.30	60347.58	418.66	74397.85	593.10	71915.91
500	524.16	63428.66	430.52	77478.93	633.16	74996.99
500	570.06	66509.74	435.38	80560.02	667.31	78078.07
500	609.27	69590.82	436.57	83641.10	689.83	81159.15
500	641.70	72671.91	434.68	86722.18	703.06	84240.24
500	667.23	75752.99	432.59	89803.26	711.55	87321.32
500	685.99	78834.07	428.97	92884.35	715.90	90402.40
500	699.15	81915.15	428.30	95965.43	718.34	93483.48
500	707.90	84996.24	423.15	99046.51	719.27	96564.57
600	70.85	41861.08	116.35	51375.36	101.69	54034.21
600	139.85	44942.16	208.53	54456.44	199.35	57115.29
600	207.10	48023.25	278.39	58142.32	289.74	60196.38
600	272.22	51104.33	325.39	62130.60	378.14	63277.46
600	335.05	54185.41	361.71	66723.69	451.87	66358.54
600	394.18	57266.49	385.55	69804.77	518.69	69439.62
600	449.92	60347.58	408.11	74397.85	579.13	72520.71
600	501.27	63428.66	420.04	78990.93	624.84	75601.79
600	547.57	66509.74	429.52	82072.02	656.53	78682.87
600	588.61	69590.82	433.69	85153.10	684.00	81763.95
600	622.70	72671.91	434.44	88234.18	698.72	84845.04
600	651.14	75752.99	433.21	91315.26	709.40	87926.12
600	673.14	78834.07	430.68	94396.35	714.30	91007.20
600	689.89	81915.15	428.19	97477.43	717.62	94088.28
600	701.61	84996.24	425.85	100558.51	718.80	97169.37
700	66.68	42314.68	113.45	51375.36	98.58	54185.41
700	131.88	45395.76	202.54	55666.04	193.66	57266.49
700	195.65	48476.85	268.55	58747.12	281.47	60347.58
700	257.62	51557.93	317.91	62886.60	366.14	63428.66
700	317.65	54639.01	348.97	65967.69	441.86	66509.74
700	374.34	57720.09	377.79	71316.77	504.76	69590.82
700	428.88	60801.18	393.22	74397.85	563.98	72671.91
700	479.51	63882.26	399.33	77478.93	613.34	75752.99
700	526.37	66963.34	401.42	80560.02	650.71	78834.07
700	567.87	70044.42	399.83	83641.10	675.13	81915.15
700	604.04	73125.51	392.64	86722.18	694.69	84996.24
700	634.90	76206.59	391.16	89803.26	706.03	88077.32
700	660.18	79287.67	388.53	92884.35	713.28	91158.40

Continued on next page

Table 3.20 – *Continued from previous page*

$n/2$	Vertical		Horizontal (d)		Horizontal (c)	
	Throughput	Cost	Throughput	Cost	Throughput	Cost
700	679.41	82368.75	386.10	95965.43	716.85	94239.48
700	693.70	85449.84	383.40	99046.51	718.66	97320.57
800	62.86	42919.48	111.24	51375.36	95.74	54185.41
800	124.81	46000.56	198.28	55212.44	186.08	57266.49
800	185.49	49081.65	263.42	59351.92	274.27	60347.58
800	244.31	52162.73	309.55	63642.60	353.24	63428.66
800	301.56	55243.81	342.90	66723.69	431.44	66509.74
800	356.57	58324.89	368.22	71316.77	496.87	69590.82
800	409.23	61405.98	381.11	74397.85	553.39	72671.91
800	459.15	64487.06	384.06	77478.93	603.86	75752.99
800	505.23	67568.14	386.09	80560.02	641.93	78834.07
800	547.55	70649.22	380.57	83641.10	670.11	81915.15
800	584.86	73730.31	376.28	86722.18	689.18	84996.24
800	617.69	76811.39	373.80	89803.26	702.06	88077.32
800	645.27	79892.47	370.12	92884.35	711.05	91158.40
800	667.24	82973.55	367.45	95965.43	715.77	94239.48
800	684.48	86054.64	369.77	99046.51	718.26	97320.57
900	60.68	42617.08	108.02	51375.36	92.45	54941.41
900	120.38	45698.16	192.08	55212.44	180.55	58022.49
900	178.55	48779.25	254.30	59956.72	264.67	61103.58
900	235.85	51860.33	301.51	63037.80	344.56	64184.66
900	291.04	54941.41	336.81	67479.69	421.66	67265.74
900	344.80	58022.49	361.24	72828.77	483.86	70346.82
900	395.95	61103.58	380.57	75909.85	541.37	73427.91
900	445.12	64184.66	390.13	78990.93	592.18	76508.99
900	490.35	67265.74	392.35	82072.02	630.79	79590.07
900	532.30	70346.82	390.47	85153.10	663.50	82671.15
900	570.63	73427.91	386.34	88234.18	684.73	85752.24
900	604.33	76508.99	386.12	91315.26	700.23	88833.32
900	632.69	79590.07	379.58	94396.35	709.38	91914.40
900	656.85	82671.15	375.86	97477.43	714.33	94995.48
900	675.77	85752.24	375.12	100558.51	717.23	98076.57
1000	57.36	41861.08	105.61	52131.36	88.61	54185.41
1000	113.33	44942.16	187.81	55212.44	176.61	57266.49
1000	168.78	48023.25	249.30	60561.52	257.61	60347.58
1000	222.85	51104.33	294.80	63642.60	335.80	63428.66
1000	275.63	54185.41	329.72	68235.69	406.62	66509.74
1000	326.35	57266.49	351.17	71316.77	473.27	69590.82
1000	375.21	60347.58	361.30	74397.85	537.55	72671.91
1000	421.65	63428.66	364.21	77478.93	580.57	75752.99
1000	466.21	66509.74	357.04	80560.02	623.20	78834.07
1000	507.25	69590.82	354.33	83641.10	653.26	81915.15
1000	545.09	72671.91	349.83	86722.18	682.03	84996.24
1000	579.53	75752.99	345.95	89803.26	697.57	88077.32
1000	610.55	78834.07	346.93	92884.35	707.67	91158.40
1000	636.36	81915.15	341.66	95965.43	713.28	94239.48
1000	658.84	84996.24	340.74	99046.51	716.71	97320.57
1100	54.76	41861.08	103.63	51677.76	85.18	54185.41
1100	108.54	44942.16	183.33	55212.44	169.89	57266.49
1100	161.47	48023.25	241.33	61166.32	251.29	60347.58

Continued on next page

Table 3.20 – *Continued from previous page*

$n/2$	Vertical		Horizontal (d)		Horizontal (c)	
	Throughput	Cost	Throughput	Cost	Throughput	Cost
1100	213.21	51104.33	292.50	64247.40	325.78	63428.66
1100	263.20	54185.41	324.02	68991.69	400.58	66509.74
1100	312.43	57266.49	345.54	72072.77	458.37	69590.82
1100	359.72	60347.58	358.05	75153.85	521.79	72671.91
1100	404.62	63428.66	358.86	78234.93	572.32	75752.99
1100	448.26	66509.74	359.70	81316.02	615.20	78834.07
1100	489.11	69590.82	351.59	84397.10	649.32	81915.15
1100	527.13	72671.91	351.35	87478.18	675.22	84996.24
1100	561.64	75752.99	346.52	90559.26	694.13	88077.32
1100	593.25	78834.07	345.35	93640.35	705.01	91158.40
1100	620.41	81915.15	346.08	96721.43	711.29	94239.48
1100	644.25	84996.24	339.59	99802.51	715.63	97320.57
1200	53.95	42617.08	100.74	52887.36	85.47	54941.41
1200	106.85	45698.16	181.42	55968.44	167.58	58022.49
1200	159.32	48779.25	239.11	60561.52	247.91	61103.58
1200	210.07	51860.33	285.33	64852.20	324.20	64184.66
1200	260.20	54941.41	317.47	68235.69	393.43	67265.74
1200	309.29	58022.49	342.87	72828.77	460.18	70346.82
1200	356.29	61103.58	356.95	75909.85	518.41	73427.91
1200	400.91	64184.66	358.78	78990.93	570.26	76508.99
1200	444.84	67265.74	361.83	82072.02	610.55	79590.07
1200	486.29	70346.82	353.98	85153.10	646.61	82671.15
1200	524.26	73427.91	352.11	88234.18	673.05	85752.24
1200	559.30	76508.99	349.29	91315.26	690.26	88833.32
1200	591.14	79590.07	347.08	94396.35	701.85	91914.40
1200	619.22	82671.15	343.40	97477.43	710.75	94995.48
1200	643.13	85752.24	344.18	100558.51	715.72	98076.57

Table 3.21: Data for the Figure 3.17 and Figure 3.18b

$n/2$	Vertical		Horizontal (d)		Horizontal (c)	
	Throughput	Cost	Throughput	Cost	Throughput	Cost
100	101.11	76861.08	132.06	95436.03	115.79	98485.74
100	201.68	79942.16	262.63	98970.71	228.28	101566.82
100	298.05	83023.25	364.03	102051.80	336.97	104647.91
100	393.72	86104.33	466.43	105132.88	448.05	107728.99
100	484.48	89185.41	538.16	108213.96	545.11	110810.07
100	575.02	92266.49	610.73	112051.04	645.66	113891.15
100	658.68	95347.58	661.60	115132.13	738.76	116972.24
100	742.30	98428.66	714.74	118969.21	828.56	120053.32
100	818.53	101509.74	746.55	122050.29	912.24	123134.40
100	894.44	104590.82	786.38	125131.37	988.11	126215.48
100	961.35	107671.91	803.90	128212.46	1061.63	129296.57
100	1029.02	110752.99	830.22	132049.54	1122.23	132377.65
100	1085.36	113834.07	845.56	135130.62	1174.85	135458.73

Continued on next page

Table 3.21 – *Continued from previous page*

$n/2$	Vertical		Horizontal (d)		Horizontal (c)	
	Throughput	Cost	Throughput	Cost	Throughput	Cost
100	1142.84	116915.15	863.95	138211.70	1233.27	138539.81
100	1188.91	119996.24	870.37	141292.79	1275.77	141620.90
200	101.11	76861.08	132.06	95436.03	115.79	98485.74
200	201.68	79942.16	262.63	98970.71	228.28	101566.82
200	298.05	83023.25	364.03	102051.80	336.97	104647.91
200	393.72	86104.33	466.43	105132.88	448.05	107728.99
200	484.48	89185.41	538.16	108213.96	545.11	110810.07
200	575.02	92266.49	610.73	112051.04	645.66	113891.15
200	658.68	95347.58	661.60	115132.13	738.76	116972.24
200	742.30	98428.66	714.74	118969.21	828.56	120053.32
200	818.53	101509.74	746.55	122050.29	912.24	123134.40
200	894.44	104590.82	786.38	125131.37	988.11	126215.48
200	961.35	107671.91	803.90	128212.46	1061.63	129296.57
200	1029.02	110752.99	830.22	132049.54	1122.23	132377.65
200	1085.36	113834.07	845.56	135130.62	1174.85	135458.73
200	1142.84	116915.15	863.95	138211.70	1233.27	138539.81
200	1188.91	119996.24	870.37	141292.79	1275.77	141620.90
300	101.11	76861.08	132.06	95436.03	115.79	98485.74
300	201.68	79942.16	262.63	98970.71	228.28	101566.82
300	298.05	83023.25	364.03	102051.80	336.97	104647.91
300	393.72	86104.33	466.43	105132.88	448.05	107728.99
300	484.48	89185.41	538.16	108213.96	545.11	110810.07
300	575.02	92266.49	610.73	112051.04	645.66	113891.15
300	658.68	95347.58	661.60	115132.13	738.76	116972.24
300	742.30	98428.66	714.74	118969.21	828.56	120053.32
300	818.53	101509.74	746.55	122050.29	912.24	123134.40
300	894.44	104590.82	786.38	125131.37	988.11	126215.48
300	961.35	107671.91	803.90	128212.46	1061.63	129296.57
300	1029.02	110752.99	830.22	132049.54	1122.23	132377.65
300	1085.36	113834.07	845.56	135130.62	1174.85	135458.73
300	1142.84	116915.15	863.95	138211.70	1233.27	138539.81
300	1188.91	119996.24	870.37	141292.79	1275.77	141620.90
400	101.11	76861.08	132.06	95436.03	115.79	98485.74
400	201.68	79942.16	262.63	98970.71	228.28	101566.82
400	298.05	83023.25	364.03	102051.80	336.97	104647.91
400	393.72	86104.33	466.43	105132.88	448.05	107728.99
400	484.48	89185.41	538.16	108213.96	545.11	110810.07
400	575.02	92266.49	610.73	112051.04	645.66	113891.15
400	658.68	95347.58	661.60	115132.13	738.76	116972.24
400	742.30	98428.66	714.74	118969.21	828.56	120053.32
400	818.53	101509.74	746.55	122050.29	912.24	123134.40
400	894.44	104590.82	786.38	125131.37	988.11	126215.48
400	961.35	107671.91	803.90	128212.46	1061.63	129296.57
400	1029.02	110752.99	830.22	132049.54	1122.23	132377.65
400	1085.36	113834.07	845.56	135130.62	1174.85	135458.73
400	1142.84	116915.15	863.95	138211.70	1233.27	138539.81
400	1188.91	119996.24	870.37	141292.79	1275.77	141620.90
500	101.11	76861.08	132.06	95436.03	115.79	98485.74
500	201.68	79942.16	262.63	98970.71	228.28	101566.82
500	298.05	83023.25	364.03	102051.80	336.97	104647.91

Continued on next page

Table 3.21 – *Continued from previous page*

$n/2$	Vertical		Horizontal (d)		Horizontal (c)	
	Throughput	Cost	Throughput	Cost	Throughput	Cost
500	393.72	86104.33	466.43	105132.88	448.05	107728.99
500	484.48	89185.41	538.16	108213.96	545.11	110810.07
500	575.02	92266.49	610.73	112051.04	645.66	113891.15
500	658.68	95347.58	661.60	115132.13	738.76	116972.24
500	742.30	98428.66	714.74	118969.21	828.56	120053.32
500	818.53	101509.74	746.55	122050.29	912.24	123134.40
500	894.44	104590.82	786.38	125131.37	988.11	126215.48
500	961.35	107671.91	803.90	128212.46	1061.63	129296.57
500	1029.02	110752.99	830.22	132049.54	1122.23	132377.65
500	1085.36	113834.07	845.56	135130.62	1174.85	135458.73
500	1142.84	116915.15	863.95	138211.70	1233.27	138539.81
500	1188.91	119996.24	870.37	141292.79	1275.77	141620.90
600	78.44	77617.08	116.67	95889.63	97.27	99241.74
600	156.54	80698.16	232.70	98970.71	198.09	102322.83
600	232.32	83779.25	322.94	102807.80	289.55	105403.91
600	307.66	86860.33	414.72	106644.88	384.52	108484.99
600	380.80	89941.41	479.78	109725.96	469.93	111566.07
600	453.15	93022.49	548.33	112807.04	559.14	114647.15
600	522.62	96103.58	594.20	116644.13	646.10	117728.24
600	592.22	99184.66	641.98	119725.21	723.01	120809.32
600	657.13	102265.74	674.89	122806.29	801.57	123890.40
600	722.80	105346.83	704.73	127399.37	875.05	126971.48
600	783.65	108427.91	734.43	130480.46	939.64	130052.57
600	844.58	111508.99	760.42	133561.54	1009.36	133133.65
600	900.93	114590.07	773.24	136642.62	1066.62	136214.73
600	957.29	117671.15	782.63	139723.70	1126.66	139295.81
600	1006.99	120752.24	796.94	144316.79	1177.87	142376.90
700	101.11	76861.08	132.06	95436.03	115.79	98485.74
700	201.68	79942.16	262.63	98970.71	228.28	101566.82
700	298.05	83023.25	364.03	102051.80	336.97	104647.91
700	393.72	86104.33	466.43	105132.88	448.05	107728.99
700	484.48	89185.41	538.16	108213.96	545.11	110810.07
700	575.02	92266.49	610.73	112051.04	645.66	113891.15
700	658.68	95347.58	661.60	115132.13	738.76	116972.24
700	742.30	98428.66	714.74	118969.21	828.56	120053.32
700	818.53	101509.74	746.55	122050.29	912.24	123134.40
700	894.44	104590.82	786.38	125131.37	988.11	126215.48
700	961.35	107671.91	803.90	128212.46	1061.63	129296.57
700	1029.02	110752.99	830.22	132049.54	1122.23	132377.65
700	1085.36	113834.07	845.56	135130.62	1174.85	135458.73
700	1142.84	116915.15	863.95	138211.70	1233.27	138539.81
700	1188.91	119996.24	870.37	141292.79	1275.77	141620.90
800	101.11	76861.08	132.06	95436.03	115.79	98485.74
800	201.68	79942.16	262.63	98970.71	228.28	101566.82
800	298.05	83023.25	364.03	102051.80	336.97	104647.91
800	393.72	86104.33	466.43	105132.88	448.05	107728.99
800	484.48	89185.41	538.16	108213.96	545.11	110810.07
800	575.02	92266.49	610.73	112051.04	645.66	113891.15
800	658.68	95347.58	661.60	115132.13	738.76	116972.24
800	742.30	98428.66	714.74	118969.21	828.56	120053.32

Continued on next page

Table 3.21 – *Continued from previous page*

$n/2$	Vertical		Horizontal (d)		Horizontal (c)	
	Throughput	Cost	Throughput	Cost	Throughput	Cost
800	818.53	101509.74	746.55	122050.29	912.24	123134.40
800	894.44	104590.82	786.38	125131.37	988.11	126215.48
800	961.35	107671.91	803.90	128212.46	1061.63	129296.57
800	1029.02	110752.99	830.22	132049.54	1122.23	132377.65
800	1085.36	113834.07	845.56	135130.62	1174.85	135458.73
800	1142.84	116915.15	863.95	138211.70	1233.27	138539.81
800	1188.91	119996.24	870.37	141292.79	1275.77	141620.90
900	66.80	79885.08	107.35	96645.63	86.44	99997.74
900	133.40	82966.16	215.87	99726.71	171.10	103078.83
900	198.81	86047.25	298.33	103563.80	255.28	106159.91
900	264.02	89128.33	384.34	106644.88	337.76	109240.99
900	327.57	92209.41	440.83	110481.96	416.68	112322.07
900	391.05	95290.49	507.69	113563.04	490.08	115403.15
900	452.63	98371.58	552.08	117551.33	569.38	118484.24
900	514.31	101452.66	600.02	120632.41	641.96	121565.32
900	573.83	104533.74	631.09	125074.29	707.18	124646.40
900	633.09	107614.83	663.59	128155.37	785.58	127727.48
900	690.16	110695.91	684.60	131236.46	842.82	130808.57
900	746.69	113776.99	713.10	135073.54	899.84	133889.65
900	800.18	116858.07	725.87	138154.62	961.32	136970.73
900	854.04	119939.15	737.27	142747.70	1008.40	140051.81
900	904.26	123020.24	753.95	145828.79	1078.25	143132.90
1000	101.11	76861.08	132.06	95436.03	115.79	98485.74
1000	201.68	79942.16	262.63	98970.71	228.28	101566.82
1000	298.05	83023.25	364.03	102051.80	336.97	104647.91
1000	393.72	86104.33	466.43	105132.88	448.05	107728.99
1000	484.48	89185.41	538.16	108213.96	545.11	110810.07
1000	575.02	92266.49	610.73	112051.04	645.66	113891.15
1000	658.68	95347.58	661.60	115132.13	738.76	116972.24
1000	742.30	98428.66	714.74	118969.21	828.56	120053.32
1000	818.53	101509.74	746.55	122050.29	912.24	123134.40
1000	894.44	104590.82	786.38	125131.37	988.11	126215.48
1000	961.35	107671.91	803.90	128212.46	1061.63	129296.57
1000	1029.02	110752.99	830.22	132049.54	1122.23	132377.65
1000	1085.36	113834.07	845.56	135130.62	1174.85	135458.73
1000	1142.84	116915.15	863.95	138211.70	1233.27	138539.81
1000	1188.91	119996.24	870.37	141292.79	1275.77	141620.90
1100	101.11	76861.08	132.06	95436.03	115.79	98485.74
1100	201.68	79942.16	262.63	98970.71	228.28	101566.82
1100	298.05	83023.25	364.03	102051.80	336.97	104647.91
1100	393.72	86104.33	466.43	105132.88	448.05	107728.99
1100	484.48	89185.41	538.16	108213.96	545.11	110810.07
1100	575.02	92266.49	610.73	112051.04	645.66	113891.15
1100	658.68	95347.58	661.60	115132.13	738.76	116972.24
1100	742.30	98428.66	714.74	118969.21	828.56	120053.32
1100	818.53	101509.74	746.55	122050.29	912.24	123134.40
1100	894.44	104590.82	786.38	125131.37	988.11	126215.48
1100	961.35	107671.91	803.90	128212.46	1061.63	129296.57
1100	1029.02	110752.99	830.22	132049.54	1122.23	132377.65
1100	1085.36	113834.07	845.56	135130.62	1174.85	135458.73

Continued on next page

Table 3.21 – *Continued from previous page*

$n/2$	Vertical		Horizontal (d)		Horizontal (c)	
	Throughput	Cost	Throughput	Cost	Throughput	Cost
1100	1142.84	116915.15	863.95	138211.70	1233.27	138539.81
1100	1188.91	119996.24	870.37	141292.79	1275.77	141620.90
1200	59.75	80338.68	101.36	97401.63	77.01	100753.74
1200	119.30	83419.76	202.50	100482.72	150.38	103834.83
1200	177.95	86500.85	279.94	104319.80	219.46	106915.91
1200	236.40	89581.93	357.41	107400.88	301.68	109996.99
1200	293.74	92663.01	418.90	110481.96	367.59	113078.07
1200	350.94	95744.09	477.25	115075.04	438.54	116159.15
1200	407.00	98825.18	520.77	118156.13	506.73	119240.24
1200	462.40	101906.26	567.66	121237.21	569.96	122321.32
1200	516.88	104987.34	597.54	125830.29	645.85	125402.40
1200	571.01	108068.43	625.83	128911.37	706.24	128483.48
1200	623.32	111149.51	647.38	131992.46	754.43	131564.57
1200	675.64	114230.59	673.89	136585.54	809.70	134645.65
1200	725.71	117311.67	689.30	139666.62	870.13	137726.73
1200	775.78	120392.75	701.01	142747.70	940.94	140807.81
1200	823.51	123473.84	712.29	145828.79	982.18	143888.90

4 Dynamic Human-Robot Collaborative Picking Strategies

4.1 Introduction

Manual picking systems have high operational flexibility, are resilient to system failures, and can handle complex situations with intuition. However, they can also experience large demand fluctuations, particularly in e-commerce, and it is challenging to meet the throughput capacity and responsiveness. Moreover, there is a shortage of human pickers in many parts of the world. These challenges have encouraged companies to invest in automated picking solutions. Some companies have even opted for fully automated warehouses. In Western Europe alone, about 40 fully automated warehouses are in operation, and many are under development (Azadeh et al., 2019a). However, these systems are very expensive and inflexible with very high setup time. These issues have given birth to robotic-based picking solutions.

In these systems, Autonomous Mobile Robots (AMR) work closely with order pickers to pick the orders. Note that AMRs differ from Automated Guided Vehicles (AGVs) as they have more degrees of autonomy. An AGV carries goods from one point to another in a warehouse on predetermined paths. The paths are pre-optimized, and it is not possible to reroute the AGVs instantaneously if unforeseen events or congestion occur. In contrast, an AMR can do an on-spot dynamic path optimization and take an alternative route if it faces any obstacles or congestion (Banker, 2018). Furthermore, AMRs use Internet of Things (IoT) principles such as machine-to-machine communication to find the safest and fastest route within the warehouse. Although AMRs are relatively cheap compared to fully automated picking systems, a large number of AMRs are required to achieve high performance, which makes the system expensive.

Pick-Support AMRs (PS-AMRs) collaborate with human pickers to carry out the order fulfillment jobs more efficiently and ergonomically. In this collaborative environment, the picker accompanies the AMR only for item picking, and the AMR autonomously carries out the remaining travel and drop off functions. Because the AMRs transport items over long distances in the warehouse, unproductive walking time of the picker reduces and the picking efficiency improves.

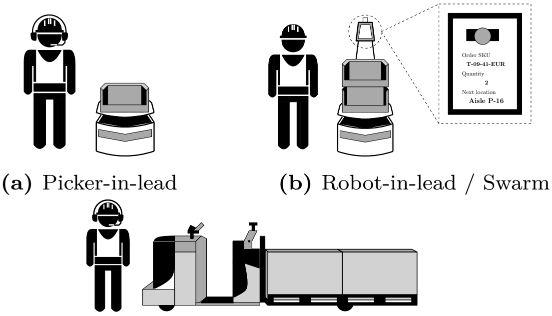
From an investment cost point of view, PS-AMRs are easily scalable. Fully manual pickers and pickers with PS-AMRs can work side-by-side. Companies can start manually, gradually automate their picking process, and still leave their current picking processes intact. Hence, investment cost and automation risk decrease significantly.

The PS-AMRs and the robots can collaborate in many different ways, but the most common are: 1) *Robot-in-lead*: the robot leads, and the picker follows the robot's instruction. A screen on top of the robot shows which item needs to be picked and where. Once all the required items are picked, the robot takes the items to the depot, and the picker is directed to another robot (e.g., by the previous robot). 2) *Picker-in-lead*: the picker leads, and the robot follows the picker. The picker receives all the required information using wearable technology (a terminal or headset). Once all items are picked, the picker sends the robots to the depot and starts picking with another robot. 3) *Swarm*: the robots and the pickers are detached from each other. Robots receive orders and travel to pick locations. A picker, who is assigned to monitor a part of the warehouse, sees the picking task on a nearby waiting robot's screen and picks the required items. The robot then travels to the next pick location, and the picker looks for another waiting robot. Once all items are picked, the robot takes the items to the depot (Trebilcock, 2018). See Figure 4.1a and Figure 4.1b for an illustration of these systems.

The described PS-AMRs are suitable for a shelf warehouse and are the focus of this study. However, there is another type of PS-AMR, known as an autonomous order picking truck, that brings similar collaboration capability to a pallet warehouse. The autonomous truck follows the picker, but the picker can also drive it to a distant location in the warehouse quickly. When the picker receives the pick information (using wearable technology), she drives the truck to the aisle from which the items need to be picked. Inside the aisle, the picker gets off the truck, and the truck autonomously starts following her along the aisle. Therefore, the picker does not need to get on and off the truck to pick each item, significantly reducing picking time. See Figure 4.1c for an illustration of such a system.

Collaborative picking systems have already been implemented by several companies, such as Fetch Robotics, Locus Robotics, Effidence, 6 River Systems, Still, Raymond, and Toyota. According to ABI Research (2019) four million robots will be installed in over 50,000 warehouses worldwide by 2025. Many of these will be PS-AMRs.

In human-robot collaborative systems, different operational policies for routing, picker, and robot allocation can be embedded in the control software. Hence, system behavior can be dynamically adjusted, which makes these systems very flexible. As a result, companies can adopt different control policies in combination with different pick strategies to improve pick performance. One of the decisions that can significantly improve the pick performance is zoning. In this strategy, the warehouse is divided into zones, with a few pickers dedicated to each zone. Zoning can reduce travel time and traffic congestion, particularly for small-sized orders and can ultimately increase system throughput (De Koster et al., 2007). Almost



(a) Picker-in-lead **(b)** Robot-in-lead / Swarm

(c) Autonomous order picking truck

Figure 4.1: A Pick-Support AMR next to a picker

all studies so far have investigated static zoning decisions (see Section 4.2 for more detail), since in traditional systems zones are hardware embedded in layout and conveyor stations. Nevertheless, when the number of lines per order is highly variable, more performance gain can be achieved by using a dynamic zoning strategy. Since human-robot collaborative systems allow fully flexible and dynamic zoning decisions, the focus of this paper is on the effect of dynamic zoning decisions in a collaborative picking system.

We investigate two strategies, the *No Zoning (NZ)* strategy and the *Progressive Zoning (PZ)* strategy. In both strategies, a pick list is assigned to a robot, and the pick locations can be located anywhere in the warehouse. In the NZ strategy, the robot is then paired with any available picker, and together they pick all the pick list items from the whole warehouse, i.e., the pickers are shared among all pick locations. Once the picking is done, the robot takes items back to the depot, and the picker becomes available for the next robot with a next pick list. In the PZ strategy, the warehouse is divided into multiple storage zones, with one or multiple pickers assigned to each zone. Pickers only pick from their dedicated zones. In every zone, the robot is paired with a picker from that zone, and together they pick all the required pick list items from that zone. If the order is incomplete, the robot progresses to another zone. Else, if all needed items are picked, it travels back to the depot, and the picker becomes available for processing the next order. The two picking operations are discussed in more detail in Section 4.3.

There are operational trade-offs involved in the selection of the two pick strategies. While the robots can travel to any pick location in both strategies, the movement of the pickers is restricted, depending on the pick strategy. In the PZ strategy, the picker's travel time is reduced because her movement is restricted to the zone. In this strategy, a partially-filled order is picked in the next zone by a different picker. However, due to demand variability among the zones, the robot may have to wait in a zone for an available picker. In the NZ strategy, the waiting time of a robot to access an available picker is reduced since pickers may access any pick location within the warehouse. However, the picker's travel time per order can increases because she may have to visit locations throughout the warehouse.

We expect that the PZ strategy has a higher throughput performance than the NZ strategy when order sizes are small because the picker's unproductive travel time is reduced. In contrast, when order sizes are large, we expect the NZ to outperform the PZ strategy since it reduces the waiting time of the robots to access an available picker. To test these hypotheses, we need to estimate the throughput capacity of the system under different pick strategies and order sizes. We use a novel queuing network model to estimate the throughput capacity of the system under a given pick strategy and order size, and compare it for the two pick strategies.

However, it is unrealistic to assume that omni-channel warehouses deal with fixed (small or large) average order sizes. They usually process various order sizes. Hence, we contest that a dynamic combination of both pick strategies can result in a better performance than operating under a fixed strategy. We are particularly interested in investigating whether the throughput capacity can be increased based on the sizes of the waiting orders by dynamic zone adjustments, without altering the number of pickers and robots. Dynamic switching between the pick strategies can be easily implemented in collaborative robot systems. For every pick tour, the pickers are instructed about the picks to be carried out (with which robot) through wearable technology or via the monitors mounted on top of the AMRs. In the context of dynamic pick strategy, we aim to answer the following research question: *Is it possible to achieve a higher pick performance with lower operational costs in a human-robot collaborative picking system, by dynamically switching between the pick strategies, given a fixed number of resources?* To find the optimal picking policy which minimizes the total operational cost, we develop a Markov Decision Process (MDP) model where the input is obtained from the queuing networks (see Figure 4.2).

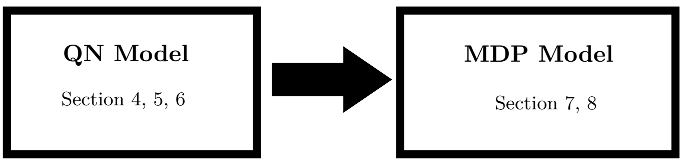


Figure 4.2: Research approach

The prime contributions of this paper are as follows: 1) we are the first to develop a stochastic model for a human-robot collaborative picking system, 2) we develop a novel closed queuing model using the technique of two-phase queuing servers (Van Doremalen, 1986) to capture the parallel movement of the picker and the robot, and 3) we are the first to investigate the effect of dynamic zoning on system performance.

The rest of the study is organized as follows. In Section 4.2, we review the related literature. In Section 4.3, we provide a detailed description of the human-robot collaborative picking operation. We then present the queuing network models for each pick strategy in Section 4.4. In Section 4.5, we present the solution approaches and estimate the performance measures of each pick strategy. In Section 4.6, we provide the numerical analysis and the obtained insights based on the queuing network models. In Section 4.7, we present the MDP framework to evaluate the performance of a dynamic order pick policy. In Section 4.8, we present the numerical analysis and the obtained insights from the comparison of different picking policies. We conclude in Section 4.9.

4.2 Literature Review

In this section, we review the literature on modeling collaborative picking systems and on the effect of zone picking decisions.

Collaborative Picking Systems: Azadeh et al. (2019a) introduced PS-AMRs and their potential for further academic research. They state that the parallel movement of pickers and AMRs makes the modeling of these systems different from manual picking and other robotic systems. Boysen et al. (2019) provide several research opportunities for the use of PS-AMRs. For instance, they suggest that traditional storage assignments, such as full turnover and class-based storage (see De Koster et al. (2007)), should be reevaluated in collaborative systems. The majority of these storage policies propose storing fast-moving items close to the depot. However, this might not be the best strategy in a collaborative environment, and different storage policies should be considered. Despite these research opportunities, to date, only a few research articles on PS-AMRs exist. Meller et al. (2018) investigate the business case for the collaborative picking systems. Given their assumption about the speed capabilities of the AMR and the pricing structure, they conclude that the business case for a collaborative system is limited to operations with low pick density. Löffler et al. (2018) study picker routing in a collaborative system. They develop an exact polynomial-time routing algorithm for a given order sequence, and different heuristic algorithms when the order sequence is a decision variable. They show that by collaborating with an AMR, a picker can reduce the walking distance by about 20 percent. Lee & Murray (2019) study a collaborative system in which two AMRs collaborate to fulfill orders: a 'picker' AMR that retrieves items from storage locations and a 'transport' AMR, which takes picked items to the depot. They define a vehicle routing problem to minimize the required time to pick all items on a pick list. Through numerical analysis, they show the system offers the greatest improvement over the traditional manual picking system when there are more parallel storage aisles or fewer cross aisles in the warehouse. The majority of the above articles model the system in a deterministic fashion, e.g., demand, travel, picking. Furthermore, Meller et al. (2018) acknowledge that the dynamics of human-robot collaboration is complex, and further research is required.

Zone Picking: Zone picking is an effective strategy to improve efficiency in both manual and automated systems. Two approaches can be used for zone picking, parallel (or synchronized) zoning, and sequential (or progressive) zoning (De Koster et al., 2007).

In *parallel (or synchronized) zone picking*, a customer order is picked simultaneously in multiple zones. Once all items are picked, they are consolidated and sorted into individual customer orders. De Koster et al. (2012) study the problem of choosing the right number of zones for a manual picking system to minimize system throughput time (picking and consolidation process). They show that for a given order size and pick list size, the throughput time is not convex in the number of zones. Roy et al. (2012) investigate the effect of the number of zones in a shuttle-based storage and retrieval system. They model the system as a semi-open queuing network. Their results suggest that having multiple zones reduces system throughput time due to shorter travel time. However, as the number of zones increases, travel time also increases due to longer waiting times for an available shuttle. Note that the additional consolidation process was not modeled in their analysis. Roy et al. (2019b) investigate the robot assignment strategies for multiple storage zones in robotic mobile fulfillment systems. They model the system as a multi-class closed queuing network. They show that the expected throughput time for order picking can be reduced by one-third by pooling robots among different zones. However, this would increase expected replenishment time by up to three times. Van Gils et al. (2018) investigate the relations between storage, batching, zone picking, and routing planning strategies. They show that all these planning decisions, including zoning, should be made in an integrated fashion.

In *sequential (or progressive) zone picking*, an order is assigned to a tote. The tote sequentially moves from one zone with storage locations to another zone, often by using a conveyor. In each zone, the required items are picked. In sequential zone picking, additional order consolidation is not required since the orders are gradually consolidated as the order tote visits each zone. De Koster (1994), Yu & De Koster (2009a, 2008), and Melacini et al. (2011) propose a queuing network model to estimate performance statistics of a conveyor-based zone picking system. In their analysis, they do not consider order tote blocking and congestion effects. However, in reality, the buffer space in each zone is limited, and therefore congestion and blocking can happen regularly in these systems. In practice, zone picking conveyor systems often use the block-and-recirculate protocol to manage the congestion dynamically. Van der Gaast et al. (2020) develop a queuing network model and propose an approximation method based on the jump-over blocking protocol (Van Dijk, 1988) to estimate the performance of a sequential zone picking system with finite buffer capacities.

In all mentioned literature, the zoning decision is static. Once the zone picking decision is made, it does not change during the picking operation. In progressive zone picking with conveyors, it is not physically possible to change the zone sizes. However, the collaborative robot system is flexible enough to work with a dynamic pick strategy. Hence, in this paper, we look at the effect of a dynamic zoning decision on the performance of a progressive zone picking system.

4.3 Description of the Pick Strategies

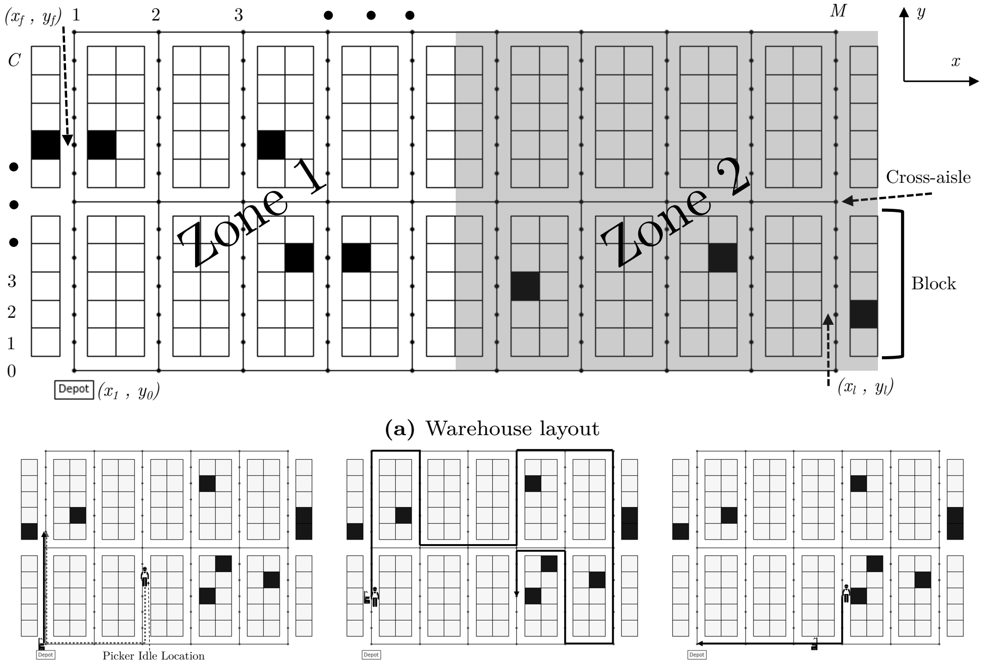
4.3.1 NZ Strategy

When an order arrives, the corresponding pick list is generated and assigned to a robot waiting in the depot (robots dwell at the depot). The robot is then paired with an available picker and travels to the first pick location. Simultaneously, the picker also travels from her dwell position to the first pick location (pickers dwell next to the last picked item of the previous pick tour). A picker's pick tour starts either from the leftmost aisle or the rightmost aisle from which an item needs to be picked. The choice depends on the dwell position of the picker. If the previous pick tour ended close to the left (right) side of the warehouse, the next pick tour begins at the leftmost (rightmost) aisle from which an item needs to be picked. This strategy minimizes the picker's travel distance and improves system performance. The picking consists of three movements: (a) A *Parallel Movement*, in which the picker and the robot with an empty pick tote travel simultaneously from their dwell positions to the first pick location, as depicted in Figure 4.3b. (b) A *Picking Items* movement, in which the picker and the robot are paired and travel to pick all the required items on the robot's pick list, as shown in Figure 4.3c. (c) A *Return to Depot* movement, in which the robot transports items to the depot, and the picker becomes available to be paired with another robot, as illustrated in Figure 4.3d. Upon completion of the last pick, the picker receives the information about the start point of the next pick tour either through wearable technology or from the monitor mounted on the previous robot.

4.3.2 PZ Strategy

In the PZ strategy, the warehouse is divided into multiple zones (see Figure 4.3a). One or multiple pickers are assigned to each zone and they only pick in their dedicated zone. Similar to the NZ strategy, robots dwell at the depot, and pickers dwell next to the location of the last picked item in their zone. When an order arrives, the corresponding pick list is generated and assigned to a robot. Then the robot travels to the first zone from which an item should be picked. The robots check and if necessary visit the zones in a sequence starting from the left (right) to the right (left) of the warehouse. In this study, we assume the depot is located in front of the leftmost aisle. Therefore, the sequence of visiting the zones starts from left to right. Zones without picks are skipped.

The first two picking movements within each zone are similar to the NZ strategy, i.e., parallel movement of the picker and the robot, as well as pairing and picking items within each zone.



(a) Warehouse layout

(b) Parallel movement **(c)** Picking items **(d)** Return to depot

Figure 4.3: Warehouse layout (a) and picking operation under the NZ strategy (b,c,d)

In the parallel movement, the robot travels from the last pick location in the previous zone to the first pick location in the current zone. Simultaneously, the picker also travels from her dwell position to the first pick location. A pick tour starts alternatively from either the leftmost aisle or the rightmost aisle of the zone from which an item needs to be picked. The choice depends on the dwell position of the picker in the zone. Once all items are picked from a zone, the robot travels to the next zone and the picker pairs with another robot. The last picking movement, i.e., return to depot, occurs upon completion of picking all items on the pick list.

In this study, we are interested in estimating the order throughput time and throughput capacity of these two pick strategies under steady-state conditions. In the next section, we show how we use queuing network models to achieve this goal.

4.4 Analytical Model

We make the following assumptions in modeling the system:

1. *Constant velocity:* We ignore acceleration and deceleration of robots and pickers. This is justified as these durations are very short. The velocity of the picker and the robot are v_p and v_r, respectively.

2. *Picker dwell position and the start of the pick tour:* In the NZ strategy, pickers dwell next to the last picked item location from the previous pick tour, and the next pick tour starts from a location relatively close to the picker's dwell position. Similarly, in the PZ strategy, pickers dwell next to the last picked item location from the previous pick tour, within their dedicated zone. The next pick tour within each zone then starts from an aisle relatively close to pickers' dwell positions, from which an item needs to be picked. This is not a limiting assumption for our analytical model but rather a good operational policy that reduces the walking distance of the pickers and ultimately increases system throughput capacity. Our model can adapt to any other policy as well.

3. *Selecting among multiple idle pickers:* In the case of multiple idle pickers, the robot is randomly paired with one of them.

4. *Pick routing:* The pick tour is S-shape or traversal (Roodbergen & De Koster, 2001) for the paired picker and robot to pick items (see Figure 4.3c). This method reduces the effect of congestion in an aisle if the pick density is sufficiently high (De Koster et al., 2007). Our model can adapt to other routing methods.

5. *Location of the depot:* The system has one depot, which is located in front of the first aisle to the left (see Figure 4.3a). Our model can adapt to include more than one depot at different locations.

6. *Two-sided picking:* Items are stored and are picked from both sides of an aisle. This is also not a limiting restriction. Goetschalckx & Ratliff (1988) have shown this is optimal in narrow aisles such as shelf racks.

7. *Uniform storage:* Items are uniformly stored in the warehouse, i.e., the probability of any location to contain a product required on an order line is based on a uniform distribution. This storage policy is appropriate for an S-shape routing and a larger pick list. Our model can adapt to any other storage policy.

8. *Sequence of checking/visiting zones:* The depot is located in front of the leftmost aisle and the sequence of visiting the zones starts from left to right. For instance, if the warehouse has three zones, the robot first checks/visits the zone on the left side of the warehouse, i.e., the zone in front of the depot, then the zone in the middle, and finally the zone on the right side. Our model can adapt to any other sequence.

9. *Congestion and blocking:* PS-AMRs can identify obstacles in their travel path and can overtake each other. Hence, congestion and blocking effects are negligible in the system.

4.4.1 Throughput Time Expression

To illustrate our model, we consider a warehouse with M aisles, three cross-aisles, and C storage columns per aisle block (see Figure 4.3a). The bold locations are those of the items to be picked with one robot, i.e., one order.

Figure 4.4 shows the flow diagram of the picking operation with a picker using the NZ strategy. Based on this flow diagram, Equation 4.1 presents the throughput time (T) of fulfilling an order.

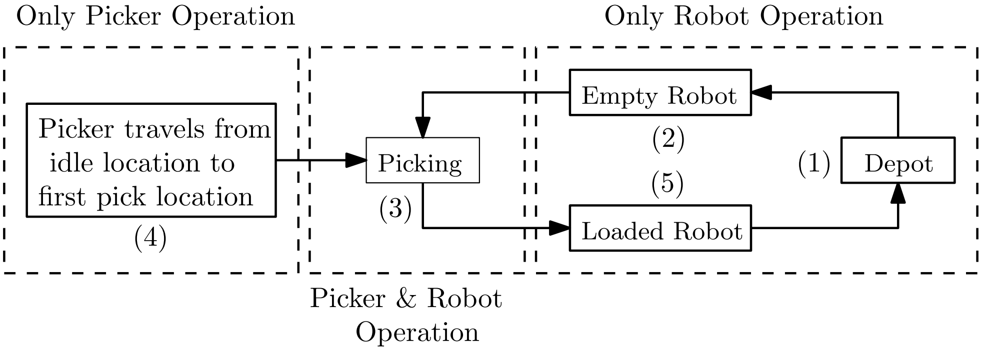


Figure 4.4: Flow diagram of a NZ strategy

$$\begin{aligned} &= \tau_d + \max\left\{\frac{x_f - x_1}{v_r} + \frac{y_f - y_0}{v_r}, \frac{|x_{P_{dwell}} - x_f|}{v_p} + \frac{|y_{P_{dwell}} - y_f|}{v_p}\right\} \\ &\quad + \tau_p + \left(\frac{x_l - x_1}{v_r} + \frac{y_l - y_0}{v_r}\right) \end{aligned} \tag{4.1}$$

The term τ_d in Equation 4.1 is the service time at the depot ((1) in Figure 4.4). The robot's travel time from the depot (x_1, y_0) to the first pick location (x_f, y_f) is described by $(x_f - x_1)/v_r + (y_f - y_0)/v_r$ ((2) in Figure 4.4). The picker's travel time to travel from her dwell position $(x_{P_{dwell}}, y_{P_{dwell}})$ to the first pick location is described by $(|x_{P_{dwell}} - x_f|)/v_p + (|y_{P_{dwell}} - y_f|)/v_p$ ((4) in Figure 4.4). Note that since the picker's dwell position and the first pick item can be located anywhere in the warehouse, the absolute value of the differences are used. The third term in Equation 4.1, τ_p, corresponds to the picking operation, in which the picker-robot pair travels to all pick locations and picks all items on the pick list ((3) in Figure 4.4). Finally the last term, $(x_l - x_1)/v_r + (y_l - y_0)/v_r$, corresponds to the robot's travel time from the last pick location (x_l, y_l) to the depot ((5) in Figure 4.4). The throughput time expression for the PZ strategy can be obtained in a similar way.

4.4.2 Queuing Network Model for NZ Strategy

To estimate the throughput capacity of the system with N robots, we model the system as a closed queuing network. In the network, the robots are the recirculating jobs, and the pickers are the servers. The picker's service time consists of two parts, traveling from the idle location to the first pick location ((4) in Figure 4.4) followed by picking items with the paired robot ((3) in Figure 4.4). Note that the picker and the robot travel from their idle location to the first pick location simultaneously, i.e., (2) and (4) in Figure 4.4 are done in parallel (see Figure 4.3b). To capture the parallel movement of the AMR and the human picker, we use a two-phase queuing server first proposed by Van Doremalen (1986) to model the picker service time. The two-phase server consists of a preparatory or setup phase, and an execution or process phase. The preparatory phase can start even when the server is idle. The execution phase can only begin when a job is waiting in the server and when the preparatory phase has been completed (Van Doremalen & Wessels, 1986). We model (4) in Figure 4.4 as the setup phase, and (3) as the process phase. We allow the setup phase to start without any jobs in the node. The process phase starts only once there is a job in the node and when the setup phase is completed, i.e., a robot is available to execute the first pick instruction jointly. We model the depot operation ((1) in Figure 4.4) as a one-phase single server (OPS) node. Since robots can easily overtake each other and congestion and blocking effects are negligible, we model the robot travel operations, i.e., (2) and (5), as infinite-server (IS) nodes. We extend the model to include more than one picker by replacing the single two-phase server node with a multi-two-phase servers (MTPS) node. Figure 4.5 illustrates the resulting queuing network.

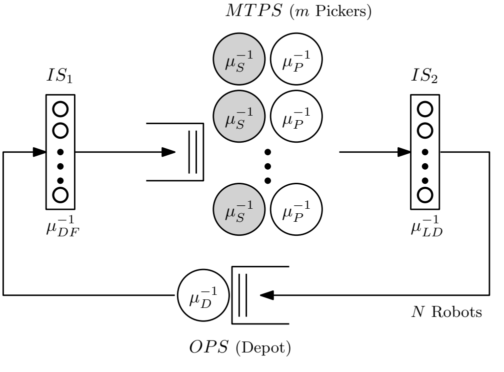


Figure 4.5: NZ strategy with m pickers and N robots (Network 1) - grey servers start their service without the presence of a robot

The expected service time at the depot is denoted by μ_D^{-1} and depends on the processing speed of robots at the depot. The expected time for the robot to travel from the depot to the first pick location is μ_{DF}^{-1}. Similarly, μ_{LD}^{-1} is the expected time of the robot to travel from the last pick location to the depot. The expected time of the picker to travel from her dwell position to the first pick location is μ_S^{-1}, and μ_P^{-1} is the expected time to travel and pick all items on the pick list. The service times μ_{DF}^{-1}, μ_{LD}^{-1}, μ_S^{-1}, μ_P^{-1} depend on the size of the assigned order to the robot, i.e., the number of items on the robot's pick list, and can be generally distributed.

4.4.3 Queuing Network Model for PZ Strategy

We extend Network 1 in Figure 4.5 to model the PZ strategy. We focus on the PZ strategy with two zones (see Figure 4.6). In Section 4.10.1, we show how the model can be extended to include multiple zones.

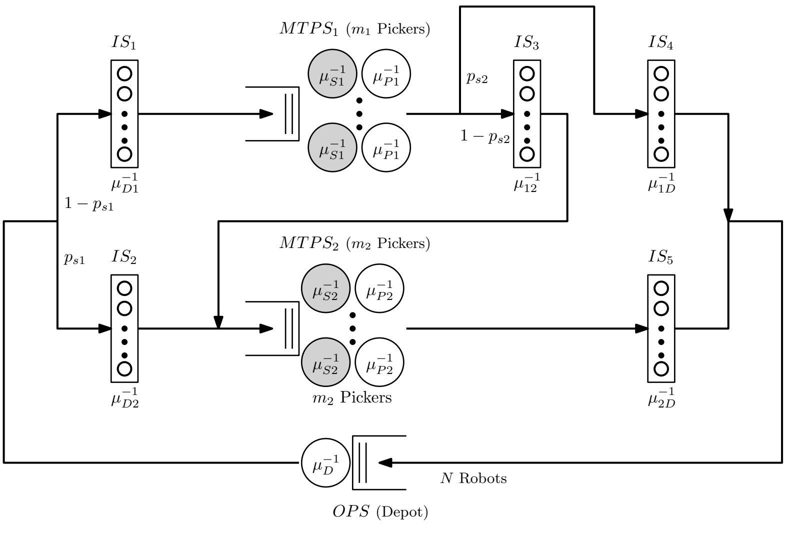


Figure 4.6: PZ strategy with two zones and N robots (Network 2) - grey servers start their service without the presence of a robot

When a robot leaves the depot, it travels to the first zone with probability $1 - p_{s1}$, or it skips the first zone and travels directly to the second zone with probability p_{s1}. Once the process in the first zone is completed, the robot moves to the second zone with probability $1-p_{s2}$, or it skips the second zone and goes to the depot with probability p_{s2}. We estimate p_{s1} and p_{s2} numerically based on the order profile. Note that similar to Network 1, the setup phase in $MTPS_1$ and $MTPS_2$ nodes (grey servers in Figure 4.6) can be done even without a robot in the node.

The expected time for the robot to travel from the depot to the first pick location in the first (second) zone is μ_{D1}^{-1} (μ_{D2}^{-1}). The expected time for the robot to travel from the last pick location in the first zone to the first pick location in the second zone is μ_{12}^{-1}. Note that the first pick location in the second zone can be on either side of the zone, depending on the dwell position of the picker. The expected time for the robot to travel from the last pick location in the first (second) zone to the depot is μ_{1D}^{-1} (μ_{2D}^{-1}). The expected setup time in zone one and zone two is given by μ_{S1}^{-1} and μ_{S2}^{-1}, respectively. Likewise, the process times in zone one and zone two are denoted by μ_{P1}^{-1} and μ_{P2}^{-1}, respectively.

4.4.4 Parameters Estimations

We use a Monte Carlo simulation to estimate the first moment of each service time, as well as the probabilities (p_s) of skipping a zone. These parameters can also be estimated analytically (see Dijkstra & Roodbergen (2017); Sadowsky & Ten Hompel (2011)). However, these procedures are elaborated and do not contribute to any additional insight towards answering our research question. Hence, we opt for the Monte Carlo simulation.

We generate 1000 instances based on input parameters and assumptions (see the beginning of Section 4.4 for the list of assumptions). The parameters include:

- Warehouse layout: including the number of aisles, number of cross-aisles, number of storage blocks, number of storage columns per block, aisle to aisle distance, and storage location width
- Resource velocity: velocity of the picker and the robot
- Order profile: number of items and their pick locations in every pick tour
- Processing time: the average time to pick an item and the average time to process a robot in the depot

For every instance, we generate a pick list and calculate the service times: 1) robot travel time from the depot to the first pick location, 2) picker travel time from her idle position to the first pick location, 3) time to pick all items, consisting of travel time and item picking time, 4) robot travel time to travel between zones in the PZ strategy, 5) robot travel time to return to depot. We then calculate the first moment of the service time for each node after running 1000 simulation instances. To calculate the skipping probabilities under the PZ strategy, we divide the number of instances that a zone is skipped by the total number of generated instances.

4.5 Solution Method for the Queuing Network Models

Since Networks 1 and 2 include two-phase servers, they do not have a product-form solution (Baskett et al., 1975; Van Doremalen & Wessels, 1986). However, the networks can still be analyzed as a continuous-time Markov chain if the state space is finite (Van Doremalen

& Wessels, 1986). However, solving the underlying Markov chain will be computationally intractable if the size of the state space is large. Fortunately, the size of the state space that describes the underlying continuous-time Markov chain for Network 1 is small and we can therefore directly analyze it to estimate the performance measures. The same procedure cannot be extended to Network 2 since the size of the state space is very large and we therefore use approximate methods to estimate performance measures. We use an aggregation-disaggregation (ADA) method and Mean Value Analysis (MVA) to estimate the performance measures (see Kumawat et al. (2018)), and assume that the service times of the nodes in the network are exponentially distributed. In Section 4.10.2, we explain how this solution approach can be extended to a network with generally distributed service times at the nodes.

4.5.1 Markov Chain Analysis of Network 1

Assuming that there are m pickers and N robots in the system, we define the state space of Network 1 by $\mathbb{S} : S = (d, r_1, r_2, s, p)$, in which d represents the number of robots in the node OPS and r_1 and r_2 are the number of robots in nodes IS_1 and IS_2, respectively. The number of active pickers in the setup phase is s, and the number of active pickers in the process phase of the node $MTPS$ is p. We always have $0 \leq d, r_1, r_2 \leq N$ and $0 \leq s+p \leq m$. Note that p depends on the number of jobs that are present in node $MTPS$ and how many pickers are busy in the setup phase. Define $i = N - (d + r_1 + r_2)$ as the number of robots in the $MTPS$ node. If $i = 0$, i.e., there is no robot in the $MTPS$ node, then $p = 0$ since the process phase can only start when there a robot is present at the node. However, the setup phase can start without a robot being present, hence, $0 \leq s \leq m$. If $i > 0$, and we have s active pickers in the setup phase, there are $m - s$ available pickers for the process phase. In that case, if $m - s \geq i$, i.e., there are more available pickers that there are robots in the $MTPS$ node, then $p = i$, and the remaining $m - s - p$ pickers stay idle and wait for another robot to enter the node. If $m - s \leq i$, i.e., there are fewer available pickers that there are robots in $MTPS$ node, then $p = m - s$ and the remaining $i - m$ robots will wait in the queue.

From state S, there are five possible transitions: 1) when a robot finishes the process at the depot, 2) when the robot finishes the process at IS_1, 3) when a picker finishes the setup phase, 4) when a picker finishes the process phase, 5) when a robot finishes the process at IS_2. Table 4.1 presents the transition rates from state $S = (d, r_1, r_2, s, p)$ to $S' = (d', r'_1, r'_2, s', p')$.

Using Table 4.1, we construct the transition rate matrix Q and obtain the steady-state probabilities numerically, using Matlab. For a reasonable number of pickers and robots, the size of the state space $\mathbb{S}$ is small and the algorithm runs very fast. For instance, for $m = 8$ and $N = 15$, the size of the state space is $|\mathbb{S}| = 7344$. We then use the steady-state

Table 4.1: State transition rates for Network 1

Type	Condition	New State	Rate
1	$d > 0$	$(d-1, r_1+1, r_2, s, p)$	μ_D
2	$s+p < m$ & $r_1 > 0$	$(d, r_1-1, r_2, s, p+1)$	$r_1\mu_{DF}$
2	$s+p = m$ & $r_1 > 0$	(d, r_1-1, r_2, s, p)	$r_1\mu_{DF}$
3	$d+r_1+r_2 = N$ & $s > 0$	$(d, r_1, r_2, s-1, p)$	$s\mu_S$
3	$0 < N-(d+r_1+r_2) < m$ & $s > 0$ & $p < N-(d+r_1+r_2)$	$(d, r_1, r_2, s-1, p+1)$	$s\mu_S$
3	$0 < N-(d+r_1+r_2) < m$ & $s > 0$ & $p = N-(d+r_1+r_2)$	$(d, r_1, r_2, s-1, p)$	$s\mu_S$
3	$N-(d+r_1+r_2) \geq m$ & $s > 0$	$(d, r_1, r_2, s-1, p+1)$	$s\mu_S$
4	$p > 0$	$(d, r_1, r_2+1, s+1, p-1)$	$p\mu_P$
5	$r_2 > 0$	$(d+1, r_1, r_2-1, s, p)$	$r_2\mu_{LD}$

probabilities to calculate all the performance metrics, such as the throughput capacity of the system with N robots.

Let $\pi(d, r_1, r_2, s, p)$ be the steady state probability of being in state (d, r_1, r_2, s, p) and let Π be the steady-state probability vector. By using Equations 4.2 and 4.3, we calculate the steady state probability of being in state (d, r_1, r_2, s, p).

$$\Pi Q = 0 \tag{4.2}$$

$$\sum_{d+r_1+r_2 \leq N} \sum_{s+p \leq m} \pi(d, r_1, r_2, s, p) = 1 \tag{4.3}$$

All the performance metrics can be calculated by using the steady-state probabilities. For instance, the throughput capacity of the system with N robots is calculated using Equation 4.4.

$$X_N = \sum_{0<d \leq N} \sum_{r_1+r_2 \leq N-d} \sum_{s+p \leq m} \pi(d, r_1, r_2, s, p)\mu_D \tag{4.4}$$

Furthermore, using the following equations, we can now calculate the average number of jobs ($Q_{O,N}$) and server utilization ($U_{O,N}$) at the OPS node and similarly the average number of jobs ($Q_{T,N}$) and server utilization ($U_{T,N}$) at the $MTPS$ node.

$$Q_{O,N} = \sum_{k=1}^{N} \left(k \sum_{r_1+r_2 \leq N-k} \sum_{s+p \leq m} \pi(k, r_1, r_2, s, p) \right) \tag{4.5}$$

$$U_{O,N} = 1 - \sum_{r_1+r_2 \leq N} \sum_{s+p \leq m} \pi(0, r_1, r_2, s, p) \tag{4.6}$$

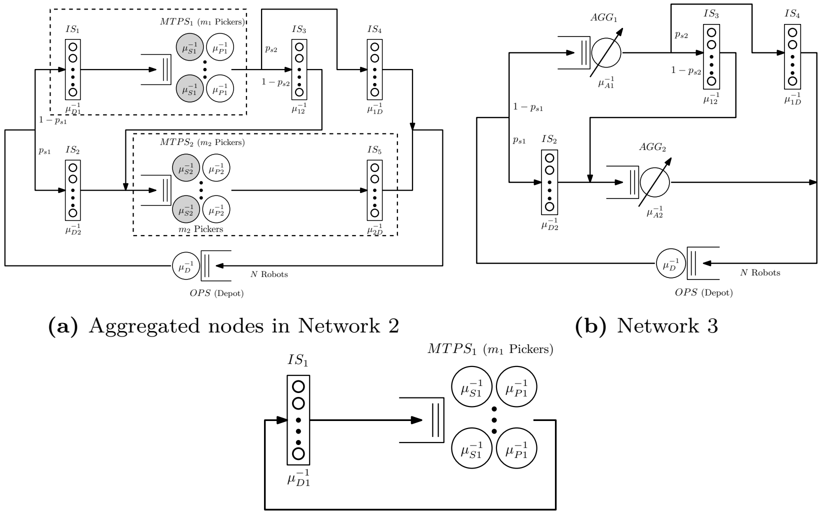
$$Q_{T,N} = \sum_{d+r_1+r_2 \leq N} \left((N-(d+r_1+r_2)) \sum_{s+p \leq m} \pi(k, r_1, r_2, s, p) \right) \tag{4.7}$$

$$U_{T,N} = \frac{X_N}{m} \left(\mu_S^{-1} + \mu_P^{-1} \right) \tag{4.8}$$

4.5.2 Aggregation Disaggregation (ADA) Based Solution for Network 2

We use an Aggregation Disaggregation (ADA) approximate method to estimate the performance of Network 2. The method is inspired by Norton's theorem for electrical circuits in which segments of the network are aggregated and replaced by flow equivalent servers with load-dependent service rates (Chandy et al., 1975; Walrand, 1983). The resulting aggregated network can be analyzed more efficiently. The disaggregation procedure reverses the aggregation process to calculate the marginal performance measures at all nodes.

Aggregation Procedure: First, we aggregate IS_1 and $MTPS_1$ and replace them with the load-dependent server AGG_1 with the mean service time of $\mu_{A1}^{-1}(n)$. Similarly, we aggregate nodes $MTPS_2$ and IS_5 and replace them with the load-dependent server AGG_2 with the mean service time of $\mu_{A2}^{-1}(n)$. We assume that AGG_1 and AGG_2 have exponential service times. Figure 4.7a illustrates which nodes are aggregated and Figure 4.7b (which we call Network 3) illustrates the resulting aggregated network.



(a) Aggregated nodes in Network 2 **(b)** Network 3

(c) Subnetwork 1

Figure 4.7: Aggregation steps

Let $n = 1, 2, ..., N$ be the number of robots in Subnetwork 1 and let n_1, n_2 be the number of robots in IS_1 and $MTPS_1$, respectively. Now we can define the state of Subnetwork 1 with (n_1, s, p), where $n_1 = 1, 2, ..., n$ and $n_2 = n - n_1$ and $s + p \leq m$. Table 4.2 presents the transition rate from state (n_1, s, p) to (n_1', s', p'). Using Table 4.2, we construct the transition rate matrix $Q_1(n)$. Let $\pi_{1,n}(n_1, s, p)$ be the steady state probability of being in state (n_1, s, p) when there are n jobs in Subnetwork 1, and let $\Pi_1(n)$ be the steady state probability vector when there are n jobs in Subnetwork 1. We calculate the steady state

Table 4.2: State transition rates for Subnetwork 1

Condition	New State	Rate
$n_1 > 0$ & $s+p < m$	$(n_1-1, s, p+1)$	$n_1\mu_{D1}$
$n_1 > 0$ & $s+p = m$	(n_1-1, s, p)	$n_1\mu_{D1}$
$n_1 = N$ & $s > 0$	$(n_1, s-1, p)$	$s\mu_S$
$N-n_1 \geq m$ & $s > 0$	$(n_1, s-1, p+1)$	$s\mu_S$
$N-n_1 < m$ & $N-n_1 > 0$ & $p < N-n_1$ & $s > 0$	$(n_1, s-1, p+1)$	$s\mu_S$
$N-n_1 < m$ & $N-n_1 > 0$ & $p = N-n_1$ & $s > 0$	$(n_1, s-1, p)$	$s\mu_S$
$p > 0$	$(n_1+1, s+1, p-1)$	$p\mu_P$

probability of being in state (n_1, s, p) by using Equations 4.9 and 4.10.

$$\Pi_1(n)Q_1(n) = 0 \tag{4.9}$$

$$\sum_{n_1 \leq n} \sum_{s+p \leq m} \pi_{1,n}(n_1, s, p) = 1 \tag{4.10}$$

Then, we calculate the throughput of Subnetwork 1, $\Lambda_1(n)$, when there are n jobs in Subnetwork 1, using Equation 4.11:

$$\Lambda_1(n) = \sum_{0 < n_1 \leq n} \sum_{s+p \leq m} \pi_{1,n}(n_1, s, p) n_1 \mu_{D1} \tag{4.11}$$

Now we replace nodes IS_1 and $MTPS_1$ with an equivalent load-dependent server AGG_1, with the mean service time $\mu_{A1}^{-1}(n) = 1/\Lambda_1(n)$ (see Figure 4.7b). Similarly we calculate $\mu_{A2}^{-1}(n)$.

Dissaggregation Procedure: Node AGG_1 is the result of aggregating nodes IS_1 and $MTPS_1$. Therefore, the number of jobs in nodes IS_1 and $MTPS_1$ is conditioned on the number of jobs in AGG_1. From the MVA procedure, we can calculate the probability of having n jobs at every node when there are N jobs circulating in the network. Let $\psi_{AGG_1,N}(n)$ be the probability of having n jobs at node AGG_1 when there are N jobs circulating in Network 3. We now calculate $P_{T1,N}(n)$, the probability of having n jobs at node $MTPS_1$ using Equation 4.12.

$$P_{T1,N}(n) = \psi_{AGG_1,N}(n) \sum_{n_1 \leq n} \sum_{s+p \leq m} \pi_{1,n}(n - n_1, s, p) \tag{4.12}$$

Next, we use $P_{T1,N}(n)$ to calculate $Q_{T1,N}$, the average number of jobs at node $MTPS_1$.

$$Q_{T1,N} = \sum_{k=1}^{N} k P_{T1,N}(k) \tag{4.13}$$

We obtain $U_{T1,N}$, the utilization of $MTPS_1$, using the following equation.

$$U_{T1,N} = \frac{V_{AGG_1} X_N}{m_1} \left(\mu_S^{-1} + \mu_P^{-1}\right) \tag{4.14}$$

Where, X_N is the throughput of the system, V_{AGG_1} is the visit ratio for node AGG_1, and m_1 is the number of servers in node $MTPS_1$.

Similarly, the number of jobs at nodes IS_5 and $MTPS_2$ is conditioned on the number of jobs at AGG_2. Therefore, the same approach can be used to estimate the performance measures for node $MTPS_2$.

4.5.3 Validation of Solution Methods

To validate the accuracy of our solution methods, we compare the analytical results with a simulation of the queuing networks. We first analyze Network 1 and Network 2 with the parameters listed in Table 4.3 and Table 4.4. All service times are assumed to be exponentially distributed. Then, using the same parameters, we build a simulation model in ARENA version 16 and obtain the performance measures again.

Table 4.3: Experiment design for validation of solution method for Network 1

Parameters	N	m	μ_D^{-1}	μ_{DF}^{-1}	μ_{LD}^{-1}	μ_S^{-1}	μ_P^{-1}
Value	2,4,6,8,10	2,4	10,15	4.3048	4.1676	3.97	25.1181

Table 4.4: Experiment design for validation of solution method for Network 2

Parameters	N	m_1	m_2	μ_D^{-1}	μ_{D1}^{-1}	μ_{D2}^{-1}	μ_{1D}^{-1}	μ_{2D}^{-1}	μ_{12}^{-1}	μ_{S1}^{-1}	μ_{P1}^{-1}	μ_{S2}^{-1}	μ_{P2}^{-1}	p_{s1}	p_{s2}
Value	2,4,6,810	1,2	1,2	10,15	2.4206	6.0524	2.3648	5.9576	5.088	4.1105	22.2418	3.9786	22.1181	0.5	0.5

For each instance, we run the simulation 50 times with a warm-up period of one hour and total simulation time of eight hours. To validate the analytical model, we measure the absolute error percentage, which is defined as $(|Y_a - Y_s|/Y_s) \times 100$, where Y_a and Y_s are performance measure estimates using analytical and simulation models, respectively. Table 4.5 and Table 4.6 summarize the absolute error percentages. The results show that the obtained results from the analytical model are reasonably accurate compared to the simulation.

Table 4.5: Absolute error percentages for Network 1

Statistics	Throughput	Utilization		Mean Number of Jobs	
		OPS	$MTPS$	OPS	$MTPS$
Mean Absolute Error(%)	0.88	0.86	1.33	2.27	10.43
Minimum Error(%)	0.06	0.00	0.05	0.05	0.40
Maximum Error(%)	4.77	5.24	3.04	13.82	15.52

Table 4.6: Absolute error percentages for Network 2

Statistics	Throughput	Utilization			Mean Number of Jobs		
		OPS	$MTPS_1$	$MTPS_2$	OPS	$MTPS_1$	$MTPS_2$
Mean Absolute Error(%)	3.04	2.96	1.94	1.44	3.93	5.46	5.01
Minimum Error(%)	0.09	0.18	0.05	0.11	0.05	0.17	0.35
Maximum Error(%)	7.05	7.21	3.78	3.49	9.24	11.52	10.45

4.6 Insights from Queuing Network Models

Business-to-Consumer (B2C) orders, such as e-commerce orders, usually consist of a single line or a relatively small number of lines per order. On the other hand, Business-to-Business (B2B) orders, such as store replenishment orders, usually include many items. In this section, we investigate the effect of order size in a multi-channel warehouse with both B2C and B2B orders, on the throughput performance of the warehouse under NZ and PZ strategies. First, we provide an asymptotic throughput analysis of the two pick strategies. Then, we set up a numerical experiment to further investigate which pick strategy results in a higher throughput depending on the order size.

In an omni-channel warehouse, we have small (B2C) and large (B2B) orders. Therefore, a dynamic order pick strategy (a dynamic switch between NZ and PZ strategies) can result in a better performance. We investigate this phenomenon in Section 4.7 and Section 4.8.

4.6.1 Asymptotic Throughput Analysis

We investigate a system with m pickers under an NZ strategy and under a PZ strategy with two equally sized zones, each with $m/2$ dedicated pickers. Furthermore, we assume that the order pick locations are uniformly spread over the warehouse, and the probability of skipping a zone in the PZ strategy is zero. Therefore, on average, half of the pick locations are located in each zone. Now, we show Proposition 1 holds.

Proposition 1. *The asymptotic (in the number of robots) throughput of a system under the NZ strategy is lower than the asymptotic throughput of the same system under the PZ strategy when the order size is small. The difference between these asymptotic throughput converges to zero as the order size increase.*

Proof. We perform a bottleneck analysis on Network 1 and Network 2 to estimate the asymptotic throughput. As the number of jobs (i.e., robots) in the network increases, the utilization of all nodes grows, and each node limits the maximum possible system throughput. Since the bottleneck node is the first to saturate, the service rate of the bottleneck node provides the upper bound to the system throughput.

We assume the depot operation is not the bottleneck. Then, the picking operation in each network will be the bottleneck, i.e., node $MTPS$ in the Network 1 and either $MTPS_1$ or $MTPS_2$ in the Network 2. Therefore, for a large number of jobs in the network, i.e., a large number of robots in the system, the asymptotic throughput of the system equals the service rate of the picking nodes. In the NZ strategy (Network 1 in Figure 4.5), the asymptotic throughput equals $X_{NZ} = m/(\mu_S^{-1} + \mu_P^{-1})$. In the PZ strategy (Network 2 in Figure 4.6), the asymptotic throughput equals $X_{PZ} = \min\{V_1(m_1/(\mu_{S1}^{-1} + \mu_{P1}^{-1})), V_2(m_2/(\mu_{S2}^{-1} + \mu_{P2}^{-1}))\}$, where V_1 and V_2 are the visit ratio of zone 1 and zone 2, respectively. The two values are the same, since 1) the probability of skipping a zone is zero, i.e., $V_1 = V_2 = 1$, 2) the number of pickers is equal in each zones, i.e., $m_1 = m_2 = m/2$, and 3) the number of items to pick in each zone is equal, i.e., $\mu_{S1}^{-1} + \mu_{P1}^{-1} = \mu_{S2}^{-1} + \mu_{P2}^{-1}$. Therefore, $X_{PZ} = (m/2)/(\mu_{S1}^{-1} + \mu_{P1}^{-1}) = (m/2)/(\mu_{S2}^{-1} + \mu_{P2}^{-1})$. Define $T_{NZ} = \mu_S^{-1} + \mu_P^{-1}$ and $T_{PZ} = \mu_{S1}^{-1} + \mu_{P1}^{-1}$. Now we need to investigate the sign of $X_{PZ} - X_{NZ}$.

$$X_{PZ} - X_{NZ} = \frac{\frac{m}{2}}{T_{PZ}} - \frac{m}{T_{NZ}} = m\left(\frac{T_{NZ} - 2T_{PZ}}{2T_{PZ}T_{NZ}}\right) \tag{4.15}$$

Since m and the denominator are positive, it is enough to investigate the sign of $T_{NZ} - 2T_{ZP}$. Equation 4.16 shows the corresponding expression (see Section 4.10.3 for the details of the derivation.)

$$T_{NZ} - 2T_{PZ} = \frac{2w}{v_p}\left(\frac{2n}{(n+1)(n+2)}\right) \tag{4.16}$$

In which, w is the distance from the front of the leftmost aisle to the front of the rightmost aisle, n is the order size, and v_p is the average picker speed. It is clear that $T_{NZ} - 2T_{PZ}$ is positive, which means X_{PZ} is higher than X_{NZ}. However, as the size of the order, n increases, $X_{PZ} - X_{NZ}$ converges to zero. □

4.6.2 Numerical Experiment

We now set up a numerical experiment to see which pick strategy has a higher throughput depending on the order size. Assume a warehouse with 20 aisles, three cross-aisles (i.e., two blocks with storage locations), and ten storage columns per aisle block (similar to Figure 4.3a). All products are randomly stored in the rack locations, and pick routes are S-shaped per block. The other parameters are listed in Table 4.7.

By using the queuing networks that we developed in Section 4.4, we calculate the throughput capacity of the system under the NZ and PZ strategies while increasing the number of robots from eight to 11 and increasing the order size from two to ten items (fixed), in steps of 1. For each combination of the order size and number of robots, we then identify the pick strategy

Table 4.7: Parameter values in the numerical experiment

Parameter	Value	Parameter	Value
# aisles	20	# cross aisles	3
# storage columns per block	20	# blocks	2
Aisle to aisle distance	3 m	Storage location width	1 m
# pickers	6	Picker velocity	0.75 m/s
# robots	8,9,10,11	Robot velocity	1 m/s
Avg item picking time	10 sec	Avg depot processing time	20 sec

that performs better and measure the improvement percentage. Figure 4.8 presents the results.

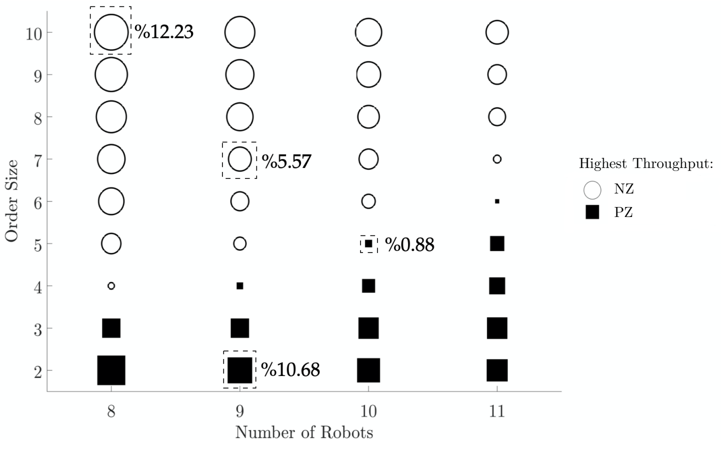


Figure 4.8: Throughput comparison between NZ and PZ strategies

The shape of the markers in Figure 4.8 illustrates which pick strategy provides a higher throughput capacity, and the size of the markers presents the magnitude of improvement compared to the other strategy. For instance, consider the system operating with eight robots and orders with ten items. In this scenario, the NZ strategy has a 12.23 percent higher throughput capacity compared to the PZ strategy. Now consider a system operating with nine robots and orders with two items. In this situation, the PZ strategy has a 10.68 percent higher throughput capacity compared to the NZ strategy. The trade-offs that we explained in the introduction can help understand the results. The throughput time (T) to fulfill an order consists of two components: waiting time and picking time. In the PZ strategy, the expected picking time is shorter since the robot transports the items from one zone to another zone. In the NZ strategy, robot waiting time is shorter since robots can pair with any available picker. In a closed queuing network, with N robots, the throughput equals N/T.

If the order size is small enough (i.e., fewer than four orders in our experiment), the expected travel distance from one zone to another is relatively long. Therefore, by using the PZ strategy and letting the robots travel this distance, a shorter throughput time and hence a higher throughput can be achieved. On the other hand, if the order size is large enough (i.e., more than seven orders in our experiment), the expected travel distance from one zone to another is relatively short. In this case, the NZ strategy has the edge over the PZ strategy, because NZ reduces the robot's waiting to find an available picker compared to PZ strategy. In other configurations, either of the pick strategies can result in better performance, depending on the order size and the number of robots in the system.

4.6.3 Insights

The NZ strategy can achieve a higher throughput performance in warehouses which process large-sized orders and is therefore suitable for the B2B channel, e.g., wholesale or the store replenishment channel. In contrast, the PZ strategy has a higher throughput performance in warehouses that process small-sized orders and is more suitable for the B2C channel, such as the e-commerce channel. However, the pick strategy that maximizes the performance in an omni-channel environment, with variable order size is unclear. Even single channel warehouses doing batch picking might have to process different batch sizes. Some batches could be small due to the required short lead time of urgent orders. Therefore, many such warehouses will face a mixture of small- and large-sized order batches.

In the next section, we develop a dynamic model to derive the optimal pick policy that minimizes the operational costs based on the number of small- and large-sized orders present in the warehouse.

4.7 Dynamic Decisions on Order Picking Strategies

In this section, we explain an MDP framework to find the optimal policy for dynamically choosing a pick strategy, which results in the lowest cost.

4.7.1 Markov Decision Process Model

We analyze a PS-AMR system that consists of a fixed number of pickers and robots. Orders arrive at an external queue with a rate of λ. An arriving order is either small or large, with a probability p or $1-p$, respectively. Since the number of pickers (m), the number of robots (N), and other operating conditions remain constant in our analysis, the fulfillment process rate depends on the order size and the pick strategy. We consider the NZ strategy and the PZ strategy with two zones and an equal number of pickers per zone. Therefore, we have four possible order processing rates: 1) small-sized orders with the NZ strategy, (μ_1^{NZ}), 2) small-sized orders with the PZ strategy (μ_1^{PZ}), 3) large-sized orders with the NZ strategy (μ_2^{NZ}), and 4) large-sized orders with the PZ strategy (μ_2^{PZ}). Figure 4.9 illustrates the resulting network with a dynamic pick strategy.

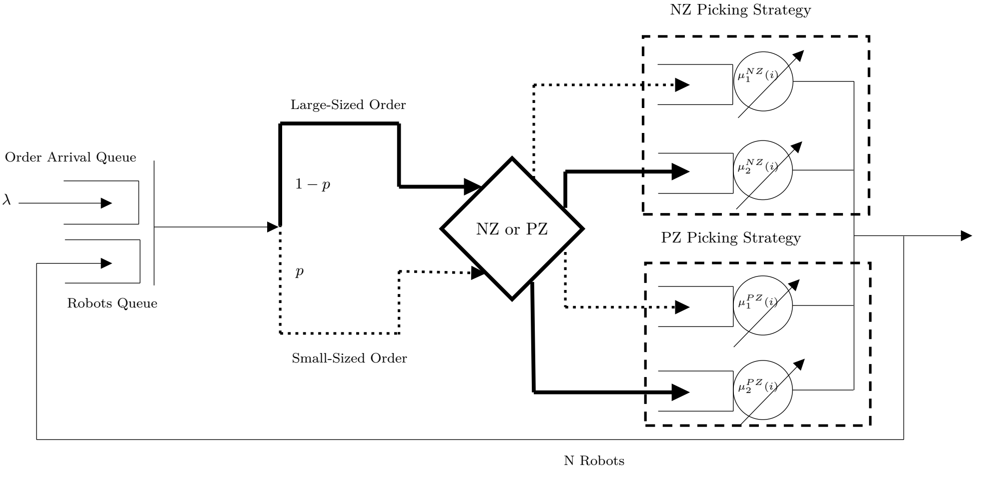


Figure 4.9: Queuing network with a dynamic pick strategy

Now, we describe the dynamics of the system as it evolves over time. The system can be described as an event-based discrete-time MDP model with process completions and order arrivals as the events. The inter-arrival and order fulfillment times are exponentially distributed. We then uniformize the decision epochs by applying uniformization technique of Lippman (1975).

State Space: Let $\mathbb{S} : s = (n_1, n_2, z)$ be the state space which describes the system. The term n_1 is the number of small-sized orders, n_2 is the number of large-sized orders present in the system, and z corresponds to the current pick strategy. If $z = 1$, the system is operating under the NZ strategy, and if $z = 2$, the current pick strategy is PZ. To ensure the system is stable, and the analysis is tractable, we assume that there is a cap, ω, on the number of orders that can be present in the system at the same time, i.e., $n_1 + n_2 \leq \omega$.

Decision Epochs: Decision epochs correspond to instances where an order arrives in the system, or an order is processed and leaves the system.

Action Space: Action space $\mathbb{A}$ consists of two actions, a, keep the current pick strategy or switch to the other pick strategy. In other words, if $z = 1$ ($z = 2$), the actions are either to keep the pick strategy, i.e., $a : z = 1$ ($a : z = 2$) or to switch to the other strategy, i.e., $a : z = 2$ ($a : z = 1$).

Depending on action a, the order processing rate equals the load-dependent throughput of one of the queuing networks developed in the previous section, i.e., Network 1 or Network 2. The term $\mu_1^a(n_1)$ represents the load-dependent throughput rate with n_1 small orders given action a. Similarly, $\mu_2^a(n_2)$ is the load-dependent throughput rate with n_2 large orders given action a. Note that if the number of orders in the system is greater than the number of robots, N, the fulfillment rate will be capped at $\mu_1^a(N)$ and $\mu_2^a(N)$. We assume switching from one strategy to the other is instantaneous. For instance, when the action is to switch from the NZ to the PZ strategy, the pickers are immediately assigned to zones, and the picking continues under the new strategy. We do not include the transition period in which some pickers have to travel from one zone to the other before the picking can continue under the new pick strategy.

Transition Probabilities: Given action a, $P^a(s,s')$ is the probability to go from state s to state s'. From state s, there are four different possible transitions: 1) a small-sized order arrival, 2) a large-sized order arrival, 3) a process completion of a small-sized order, and 4) a process completion of a large-sized order. Table 4.8 presents the transition from state $s=(n_1,n_2,z)$ to $s'=(n',n',z')$.

Table 4.8: Possible transitions from state $s=(n_1,n_2,z)$ and action $a \in \mathbb{A}$

	Type	Condition(s)	New State	Transition Probability $P^a(s,s')$
	1	$0 \le n_1,\ 0 \le n_1+n_2 < \omega$	(n_1+1, n_2, z^a)	$\frac{p\lambda}{\nu}$
$(*)$	1	$n_1+n_2=\omega$	(n_1, n_2, z^a)	$\frac{p\lambda}{\nu}$
	2	$0 \le n_2,\ 0 \le n_1+n_2 < \omega$	(n_1, n_2+1, z^a)	$\frac{(1-p)\lambda}{\nu}$
$(*)$	2	$n_1+n_2=\omega$	(n_1, n_2, z^a)	$\frac{(1-p)\lambda}{\nu}$
	3	$0 < n_1,\ n_1+n_2 \le N,$	(n_1-1, n_2, z^a)	$\frac{\mu_1^a(n_1)}{\nu}$
$(**)$	3	$0 < n_1,\ n_1+n_2 > N,$	(n_1-1, n_2, z^a)	$\frac{\mu_1^a(N_{n_1})}{\nu}$
	4	$0 < n_2,\ n_1+n_2 \le N,$	(n_1, n_2-1, z^a)	$\frac{\mu_2^a(n_2)}{\nu}$
$(**)$	4	$0 < n_2,\ n_1+n_2 > N,$	(n_1, n_2-1, z^a)	$\frac{\mu_2^a(N_{n_2})}{\nu}$
$(***)$	-	-	(n_1, n_2, z^a)	$1-\gamma$

For states in which the number of present orders in the system equals ω, i.e., scenarios which are denoted by $(*)$ in Table 4.8, arrival orders are rejected (in reality these orders would be postponed). Therefore, n_1 and n_2 do not change. For states in which the number of orders is greater than the number of robots, $n_1+n_2 > N$, we need to decide how many robots are processing small-sized and how many are processing large-sized orders. In these scenarios, which are denoted by $(**)$ in Table 4.8, we assume N_{n_1} robots are processing small-sized and N_{n_2} robots are processing large-sized orders. These values are proportional to the number of small- and large-sized orders in the system, i.e., $N_{n_1} = \max\{1, \lfloor Nn_1/(n_1+n_2)\rfloor\}$ and $N_{n_2} = N - N_{n_1}$. We also set the uniformization rate $\nu = \lambda + \max\{\mu_1^{NZ}(N), \mu_1^{PZ}(N)\} + \max\{\mu_2^{NZ}(N), \mu_2^{PZ}(N)\}$. The last scenario in Table 4.8, which is denoted by $(***)$, is the result of the uniformization of the decision epochs. It means that we stay in the same state with probability $1-\gamma$, in which γ is the sum of all other possible transition probabilities from the state $s=(n_1,n_2,z)$.

Cost Function: The cost function consists of three components: 1) *Order fulfilment cost*, which is the cost of processing the orders. 2) *Order postponement cost*, which is the cost of not being able to fulfill an incoming order, i.e., when $n_1 + n_2 = \omega$. 3) *Order pick strategy switching cost*, which is the cost associated with switching the order pick strategy. Let C_w be the order fulfillment cost per order, C_p be the postponement cost per unit time per order, and C_s be the pick strategy switching cost per time unit. Therefore, the total cost g per time unit to be in state s after decision a is taken is given by the sum of all three cost components and is defined as follows:

$$g(s,a) = (n_1 + n_2)C_w + \mathbb{1}^s_{[n_1+n_2=\omega]}\lambda C_p + \mathbb{1}^s_{[z^a \neq z]}C_s \tag{4.17}$$

In which $\mathbb{1}^s_{[n_1+n_2=\omega]}$ is an indicator function, which equals one if we are in state s where $n_1 + n_2 = \omega$ and zero otherwise. Similarly, $\mathbb{1}^s_{[z^a \neq z]}$ is an indicator function, which equals one, if the action in state s is to change the pick strategy, and zero otherwise.

4.7.2 Solving the MDP Model

The state transitions in the system, i.e. an arrival or a departure of an order, happen quickly. Therefore, decisions are made frequently, and the discount factor of future costs is very close to one. Consequently, we compare different policies based on the average expected cost criterion (Puterman, 2014).

Without loss of generality, we assume that the uniformization rate, ν, equals one. We can achieve this by scaling the event and cost rates accordingly. As a result, the total expected cost between two consecutive events for a given state s and action a will equal $g(s,a)$.

Now, we define a stationary policy f as a function from $\mathbb{S} \to \mathbb{A}$ that describes which action a to choose, for every state s, once a transition occurs. For any state s, let π^f_s be the stationary probability to be in state s under stationary policy f. If a unique π^f_s exists, then the average cost per time unit under the stationary policy f, C^f_{avg} is calculated by the following equation:

$$C^f_{avg} = \sum_{s \in \mathbb{S}} \pi^f_s g(s, f(s)) \tag{4.18}$$

And the optimal stationary policy f^* is a policy that minimizes C^f_{avg}.

$$f^* = \operatorname*{argmin}_f C^f_{avg} \tag{4.19}$$

Note that the optimal average cost per time unit, C^f_{avg}, is independent of the starting state. In Proposition 2, we show that the optimal stationary policy exists, and the minimum average cost per time period is finite.

Proposition 2. *A stationary policy f^* exists (as well as unique stationary probabilities $\pi_s^{f^*}$) which leads to a finite minimum average cost per time period, C^*_{avg}.*

Proof. Cavazos-Cadena & Sennott (1992) show that if:

a) there exists a stationary policy which induces an irreducible, positive recurrent Markov Chain with finite average cost, and

b) for each positive number M, the number of states for which the average cost is less than M is finite,

then, the existence of an average-cost-optimal stationary policy with the finite average cost is guaranteed.

To prove (a), assume the following stationary policy f, in which we always stick with the current pick strategy except for two states. If we are in state $(0, 0, 1)$, meaning there are no orders in the system and the current pick strategy NZ, and a small-sized order arrives, the action is to change from NZ to PZ and move to state $(1, 0, 2)$. In the other case, if we are in state $(0, 0, 2)$, meaning there are no orders in the system and the current pick strategy is PZ, and a large-sized order arrives, the action is to change form PZ to NZ and move to state $(0, 1, 1)$. This policy, f, induces an irreducible Markov chain. To verify this property, we divided the state space into two subsets $\mathbb{S} = \mathbb{S}_1 \cup \mathbb{S}_2$ where, $\mathbb{S}_1 : s_1 = (n_1, n_2, 1)$ and $\mathbb{S}_2 : s_2 = (n_1, n_2, 2)$. Under policy f, it is possible to reach any state from any other state within subsets $\mathbb{S}_1$ and $\mathbb{S}_2$. This can be verified from Table 4.8. Moving from $\mathbb{S}_1$ to $\mathbb{S}_2$ or vice versa is possible through state $(0, 0, 1)$ and $(0, 0, 2)$, respectively. With probability $p\lambda/\nu$ the action is to go from state $(0, 0, 1)$ to state $(1, 0, 2)$, i.e. the state moves from $\mathbb{S}_1$ to $\mathbb{S}_2$. Furthermore, with probability $(1 - p)\lambda/\nu$ the action is to go from state $(0, 0, 2)$ to state $(0, 1, 1)$, i.e. the state moves from $\mathbb{S}_2$ to $\mathbb{S}_1$. Therefore, any state in $\mathbb{S}$ can be reached from any other state and therefore, the induced Markov chain is irreducible. It is well-known that every irreducible Markov chain with a finite state space is positive recurrent (Karl, 2009). Since the number of states in our model is finite ($n_1 + n_2 \le \omega$ and $z = 1, 2$ therefore, $|\mathbb{S}| = \omega(\omega + 1)$), therefore, the induced Markov chain by policy f is positive recurrent. Hence, policy f induces an irreducible positive recurrent Markov chain and a unique π^f exists. Furthermore, the average cost of the policy, C^f_{avg} is also finite since the number of states is finite and all the components of the cost function are also finite:

$$\begin{aligned} C^f_{avg} &= \sum_{s\in\mathbb{S}} \pi^f_s g(s, f(s)) \\ &= \sum_{s\in\mathbb{S}} \pi^f_s \left((n_1 + n_2)C_w + \mathbb{1}^s_{[n_1+n_2=\omega]}\lambda C_p + \mathbb{1}_{[z^a \neq z]} C_s \right) \\ &\le \sum_{s\in\mathbb{S}} \pi^f_s \left(\omega C_w + \lambda C_p + C_s \right) < \infty \end{aligned} \tag{4.20}$$

To prove (b), since the number of states is finite, and for every state the cost is also finite, for every positive number L the number of states for which $g(s,a) \leq L$ is finite. Therefore, there exists a stationary policy that leads to a finite minimum average cost per time period. □

Dynamic programming can be used to obtain the optimal policy. We first need to define $h^*(s)$ a relative or differential cost function for each state s. Let $\hat{s} \in \mathbb{S}$ be a recurrent state, then $h^*(s)$ is the minimum of the difference between the expected cost to reach $\hat{s}$ from s for the first time and the cost to reach $\hat{s}$ from s for the first time if we incurred the C^*_{avg} on every step (Bertsekas, 1995). Equation 4.21 presents the Bellman equation corresponding to our MDP model.

$$C^*_{avg} + h^*(s) = \min_{f(s)\in\mathbb{A}} \left(g(s, f(s)) + \sum_{s'\in\mathbb{S}} P^{f(s)}(s,s')h^*(s') \right) \forall s \in \mathbb{S} \tag{4.21}$$

Where C^*_{avg} is the optimal average cost per time period and $h^*(\hat{s}) = 0$ for a specific recurrent state $\hat{s} \in \mathbb{S}$. Any of the common solution algorithms (such as value iteration or policy iteration) can be used to determine the optimal policy (Puterman, 2014).

4.8 Numerical Analysis and Obtained Insights

We assume a warehouse layout similar to the one presented in Figure 4.3a. The warehouse contains 20 aisles. Each aisle is divided into two storage blocks, i.e., the warehouse has three cross-aisles. Each aisle in each block has 20 picking positions on either side. Therefore, there are 1600 pick positions in total. The length of a pick position and the width of a cross-aisle is one meter. The distance between neighboring aisles is three meters. This type of layout represents a typical shelf warehouse with aisles that are sufficiently wide to allow robots to overtake. The depot is located in front of the leftmost aisle. There are six pickers in the warehouse, and we vary the number of robots between eight and 11.

We use order data from one of the warehouses of a logistics service company in the Netherlands, with a similar warehouse layout, that fulfills both e-commerce and store replenishment orders for a non-food store chain retailer (see Section 4.10.4 for detailed analysis of the data). We distinguish the small- and large-sized orders and their frequency of occurrence based on these data. Orders with five or fewer items (with an average of 1.8 order lines per order) are classified as small-sized orders (the e-commerce channel) and orders with five or more items (with an average of 35.3 order lines per order) are considered as large-sized orders (the store channel). The average cost of postponing an order to the next day can be estimated as $C_p = 20$ € per order. If an order requires more than 24 hours to process, we assume it is postponed to the next day. Therefore, the incremental order fulfillment cost can be estimated at $C_w = 0.00024$ € per order per second. The switching cost is set to $C_s =$

0.001 € every time a switch is made from one strategy to the other. The maximum number of orders that can be present simultaneously in the system, ω, equals 20. The average order arrival rate is assumed to be $\lambda = 200$ orders per hour. The probability of small-sized order, p, varies from 0.05 to 0.95 with a step size of 0.05. Table 4.9 presents all the parameters in the experiment.

Table 4.9: Parameter values in the numerical experiment

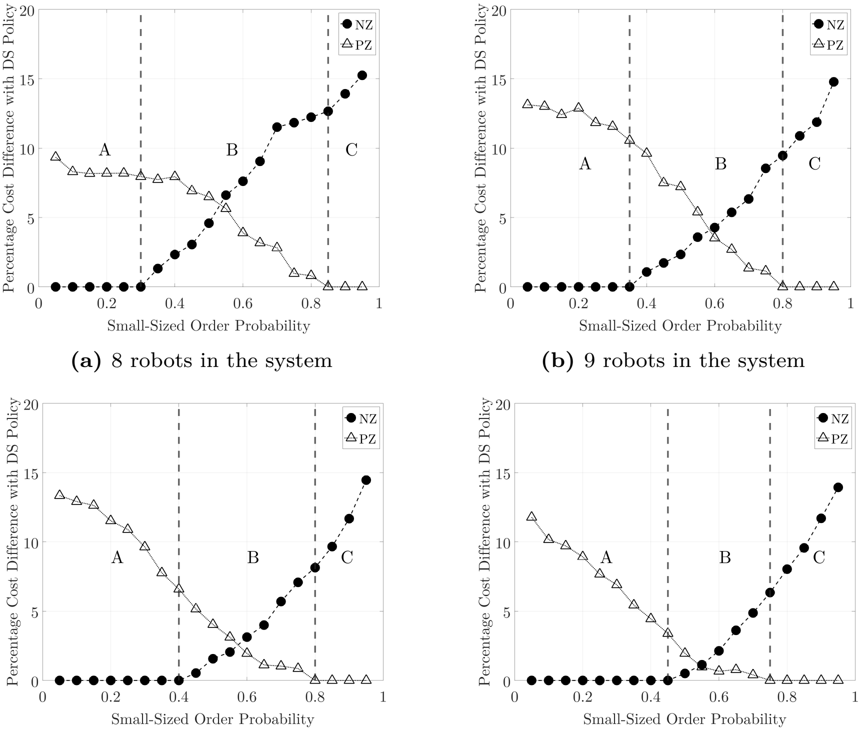
Parameter	Value	Parameter	Value
# aisles	20	# middle X-aisles	1
# pick position per block	20	# storage block	2
Aisle to aisle distance	3 m	Picking position length	1m
Avg item picking time	10 sec	Avg depot processing time	20 sec
# pickers	6	Picker velocity	0.75 m/s
# robots	8,9,10,11	Robot velocity	1 m/s
Postponement cost	20 €/order	Order processing cost	0.00024 €/order.sec
Switching cost	0.001 €		
Average arrival rate	200 orders/hour	Probability of small-sized order	0.05-0.95 (step size: 0.05)

4.8.1 Dynamic Switching Policy Based on the Number of Orders in the System

For each scenario, we use the open-source MDP toolbox for Matlab from INRA to determine the optimal Dynamic Switching (DS) policy, f^{DS} (Chadès et al., 2014). This policy is based on the number of small- and large-sized orders that are simultaneously present in the system. We then compare the performance of the DS policy and the fixed NZ or PZ policies, i.e., f^{NZ} and f^{PZ}, which corresponds to NZ and PZ strategy, respectively.

We use a discrete event simulation of the corresponding Markov Decision Process under each of the policies f, to calculate the average cost, C^{f}_{avg}, per order per time unit (we simulate order arrivals in the network shown in Figure 4.9). The pick strategy in the simulation is chosen based on policy f. For each scenario, we simulate one week of operation, and we run the simulation ten times and record the 95 percent confidence interval of the average cost per order per time unit. For the scenarios in which the confidence intervals under different policies do not overlap, we calculate the percentage gap between the average cost of operating under the DS policy and under each of the fixed NZ and PZ policies. For instance the gap between the NZ and the DS policy is calculated using $\left(C^{f^{NZ}}_{avg} - C^{f^{DS}}_{avg}\right)/C^{f^{DS}}_{avg} \times 100$. The gap between the PZ and the DS policy is calculated in the same way. Figure 4.10 presents the results.

Each of the graphs is divided into three regions, A, B, and C. In region A, NZ is the optimal policy, and the average cost is up to 15 percent lower compared to the PZ policy. This region represents a warehouse which receives mostly large-sized orders, i.e., a store replenishment warehouse. In region C, PZ is the optimal policy, and average cost is 15 percent lower



(a) 8 robots in the system

(b) 9 robots in the system

(c) 10 robots in the system

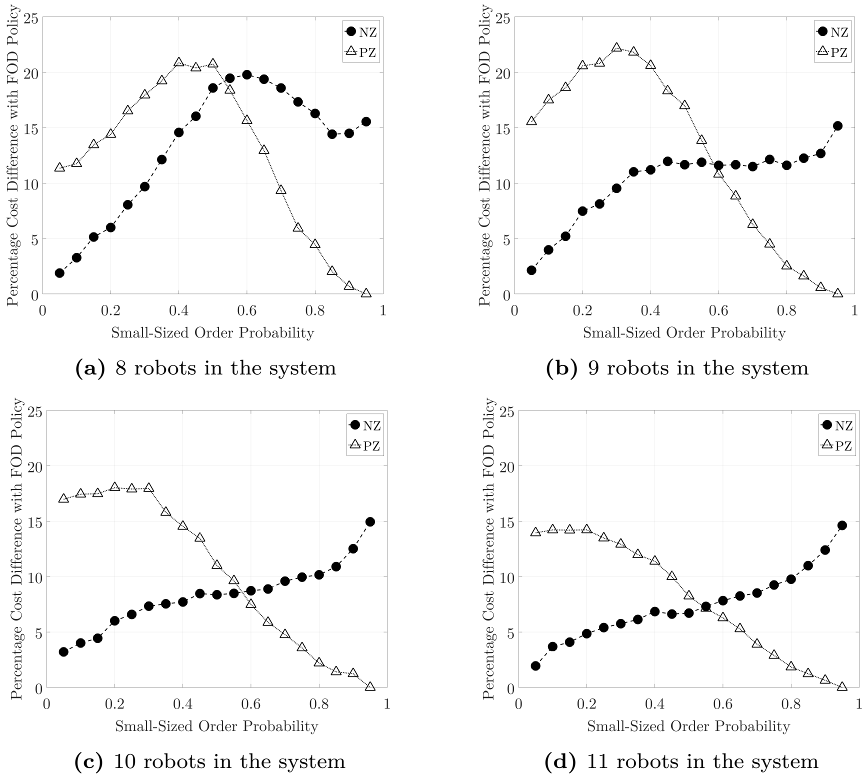
(d) 11 robots in the system

Figure 4.10: Percentage gap between DS and NZ/PZ policies

compared to NZ policy. This region represents a warehouse which receives mostly small-sized orders, i.e., an e-commerce warehouse. These observations are in line with the insights from the trade-off analysis shown in Section 4.6. In region B, DS is the optimal policy, and the average cost is seven percent lower that the two fixed policies. In this region, around 40 to 80 percent of the orders are small-sized, and the rest are large-sized orders, i.e., an omni-channel warehouse. However, region B shrinks as the number of operating robots increases. In other words, the largest performance gain of using the DS policy is obtained with a limited number of operating robots.

4.8.2 Fixed Order Size Dependent Policy

In this section, a policy is explored in which both pick strategies are used at the same time. In other words, we use the PZ strategy for all small-sized orders and the NZ strategy for all large-sized orders. We run the simulation using the same parameters as before under this Fixed Order size Dependent (FOD) policy. Similarly to the previous section, we calculate the percentage gap between the average cost of operating under the FOD policy and each of the fixed NZ and PZ policies. Figure 4.11 presents the results.



(a) 8 robots in the system

(b) 9 robots in the system

(c) 10 robots in the system

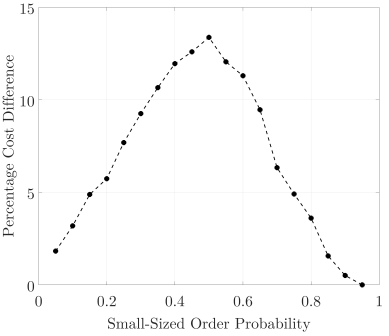
(d) 11 robots in the system

Figure 4.11: Percentage gap between FOD and NZ/PZ policies

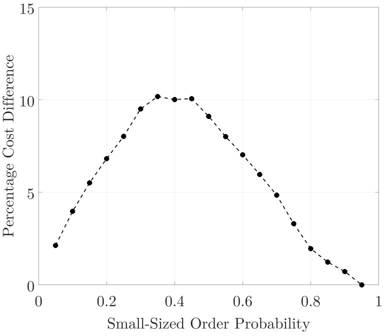
Figure 4.11 shows that the FOD policy provides the lowest average cost since it benefits from both pick strategies. Note that a region A (where the NZ policy is optimal) does not exist, and the region C (where the PZ policy is optimal) is very small, i.e., only when more than 95 percent of the incoming orders are small-sized. Therefore, in the majority of the scenarios, the FOD policy outperforms the fixed policies by up to 19 percent. Similar to the DS policy, the performance gain of FOD policy compared to the fixed policies is more pronounced when the number of robots is small. Figure 4.12 presents the gap between the FOD and DS policies, i.e., $\left(C_{avg}^{f^{DS}} - C_{avg}^{f^{FOD}}\right)/C_{avg}^{f^{FOD}} \times 100$.

4.8.3 Discussion

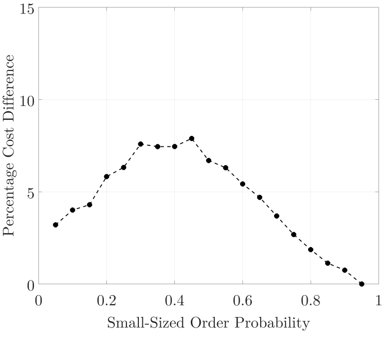
The FOD policy should be treated as a benchmark to showcase the potential performance gain of this system. Although it performs better than the NZ or PZ policies (see Figure 4.11



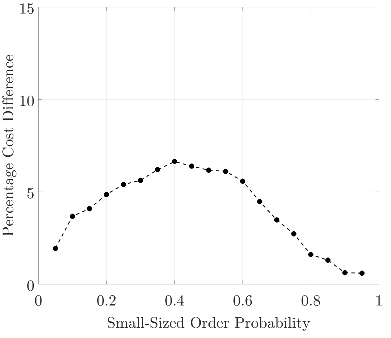
(a) 8 robots in the system



(b) 9 robots in the system



(c) 10 robots in the system



(d) 11 robots in the system

Figure 4.12: Percentage gap between FOD policy and DS policy

and Figure 4.12), the FOD policy may be hard to implement in practice. In the FOD policy, the picker might need to frequently change the pick strategy because she needs to adapt to the size of the incoming order. Furthermore, if you have a small order at some point, it may take a while before another picker is available to pick it up in the other zone. Moreover, the constant change can be confusing for pickers and ultimately may affect their pick performance. On the other hand, strategy switching happens less frequently in the DS policy due to the switching cost.

A second implementation problem of the FOD policy may be the perceived workload fairness of the pickers. We know that pickers walk less under the PZ strategy compared to the NZ strategy. In the FOD policy, some pickers might end up picking more orders under the NZ strategy compared to others. This may create a perception among these workers that they are traveling longer distances compared to those picking under PZ strategy. This problem does not exist in the DS policy since all pickers work either under the NZ or PZ strategy all the time.

4.8.4 Insights

When a warehouse operates with a large number of robots per picker (more than two in our numerical analysis), a fixed policy is the optimal or near-optimal policy depending on the order size. In other words, no substantial cost savings can be achieved by adopting a DS policy. However, with a small number of robots per picker (around 1.3 in our numerical analysis) the DS policy can reduce costs by up to 7 percent when 55 percent of incoming orders are small. This is particularly important for companies that want to automate the picking operation gradually. They can start with a lower number of robots and operate under the DS policy. Once the robot fleet has increased to a certain size, a fixed policy can be deployed.

The PS-AMR system can increase performance by using FOD policy, where both pick strategies are utilized at the same time. However, this policy may not be feasible in practice.

4.9 Conclusions

In this research, we investigate the Pick-Support AMR system, where robots collaborate with human pickers to fulfill the orders. This human-robot collaborative system is flexible enough to adopt different pick strategies. We investigate the effect of zoning the warehouse on the performance of the system. We focus on two pick strategies. The NZ strategy reduces robot waiting time as the pickers are pooled, whereas the PZ strategy reduces the picker travel time by using robots to transport the partially filled orders from one zone to the other. We show that the average order size affects the choice of the zoning strategy. A novel queuing network model is developed to estimate the system's performance under different pick strategies and order sizes to test our conjectures. Particularly, a two-phase queuing process is used to capture the realistic simultaneous movement of the robots and the pickers in the system. The results show that the NZ (PZ) strategy results in a higher throughput performance when the average order size is large (small). Therefore, the PZ strategy is useful in e-commerce warehouses that predominantly cater for small order sizes, and the NZ strategy is suitable for store replenishment warehouses that predominantly cater for large order sizes.

However, orders in many warehouses (e.g., omni-channel) are heterogeneous in terms of order size. As a result, a fixed picking strategy might not be the best pick strategy. Hence, we propose a DS policy based on the number of small- and large-sized orders in the system and compare its performance, in terms of cost, with the other two fixed policies, i.e., NZ and PZ. Our results show when a warehouse is operating with a large number of robots per picker (about two), the DS policy does not generate substantial operational cost savings. However, when the number of robots per picker is limited (about 1.3), operational costs decrease by up to seven percent. This result is particularly useful for companies that are interested in gradually automating their warehousing operations.

We also investigated an FOD policy, in which both pick strategies are used at the same time. The FOD policy reduces operational cost by up to 19 percents compared to fixed policies. However, this policy should primarily be seen as a benchmark to show the potential performance gain of this system, as it may be difficult to use in practice.

4.10 Appendix

4.10.1 Queuing Network Model with Multiple Zones

In this section, we show how we can extend the model to incorporate multiple zones. To do this, we need to have a model that can be systematically extended to multiple zones, we need to make additional assumptions on how a robot visits the zones. For every zone, we defined the entry and exit points. The entry point is located in front of the leftmost aisle in the zone, and the exit point is located in front of the rightmost aisle in the zone (see Figure 4.13.)

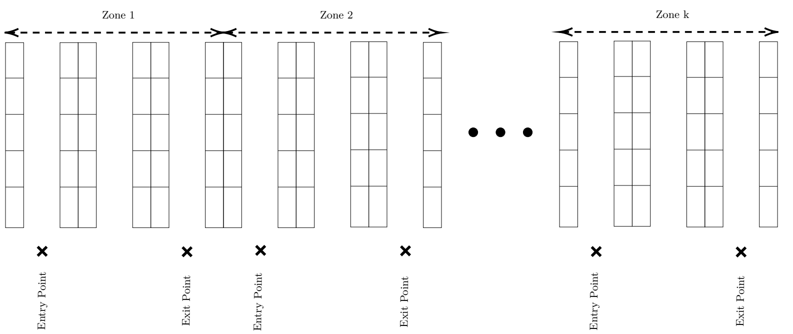


Figure 4.13: Enter and exit points for each zone

At the entry point of each zone, the robot checks whether it needs to visit that zone or not. If the zone needs to be skipped, the robot travels to the entry point of the next zone. If there are items to pick in the zone, the robot enters the zone, and once all items from the zones are picked, it travels to the exit point of the zone. There, the robot checks whether all items are picked or not. If so, it brings back items to the depot; if not, it continues to the next zone. Figure 4.14 presents the resulting queuing network model with k zones and N robots.

The expected time for the robot to travel from the depot to the entry point of zone one is $\mu_{DE_1}^{-1}$. The expected time for the robot to travel from the entry point of zone i to the

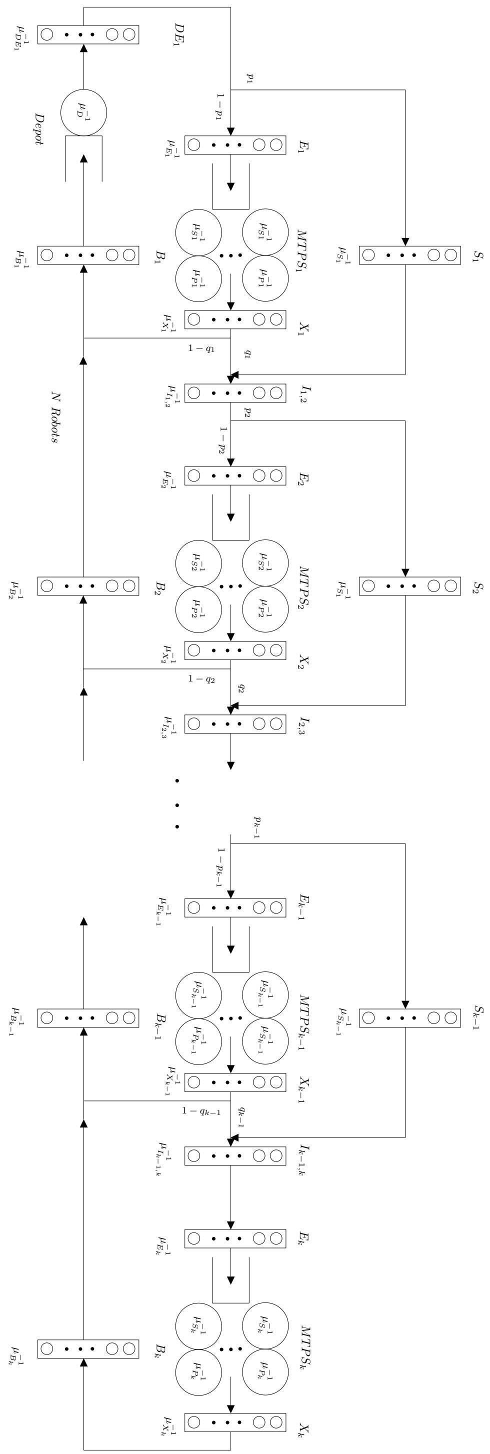


Figure 4.14: Queuing network of PZ strategy with k zones

first pick location within zone i is $\mu_{E_i}^{-1}$. The expected time for the robot to travel from the last pick location in zone i to the exit point of zone i is $\mu_{X_i}^{-1}$. The expected setup and process time in zone i are given by $\mu_{S_i}^{-1}$ and $\mu_{P_i}^{-1}$, respectively. The expected time for the robot to travel from the exit point of zone $i-1$ to the entry point of zone i is $\mu_{I_{i-1,i}}^{-1}$. The probability to skip zone i is denoted by p_i. When zone i is skipped, the robot travels from the entry point of zone i to the entry point of zone $i+1$ with an expected travel time of $\mu_{S_i}^{-1}$. All items are picked after visiting zone i with probability $1-q_i$. The expected time of the robot to travel from the exit point of zone i to the exit point of zone $i-1$, when $i \neq 1$, is $\mu_{B_i}^{-1}$. The expected time of the robot to travel from the exit point of zone one to the depot is $\mu_{B_1}^{-1}$. We can use a combination of an aggregation disaggregation method and the mean value analysis to estimate the performance measure of this network.

4.10.2 Network with Generally Distributed Service Time Nodes

By approximating the generally distributed service times with a phase-type Erlang distribution, we can solve networks with generally distributed service time nodes. For instance, let Y be the service time for a node. Y is a random variable with a mean $E[Y]$ and a squared coefficient of variation CV_Y^2. If $CV_Y^2 > 0$, Y can be approximated by an Erlang distribution $Er(u,k)$ in which $k = \lceil \frac{1}{CV_Y^2} \rceil$, and $u = \frac{k}{E[Y]}$. By using an Erlang distribution to approximate general service time, we can describe the system using a Markov chain with a finite state space. Then, a method similar to the one presented in Section 4.5.1 and Section 4.5.2 can be used to obtain the network performance measures.

4.10.3 Analytical Expression of Picker Expected Travel Time under NZ and PZ Strategies

Most of the analysis is based on Sadowsky & Ten Hompel (2011). Let x be the distance from the front of the leftmost aisle to the front of the aisle where an item needs to be picked. Now, we define an aisle access frequency as the average number of times a picker has to travel to an aisle over a certain period of time. Figure 4.15 shows an empirical distribution of the aisle access frequency over x.

Given items are uniformly stored in the warehouse, we can approximately model the empirical distribution of x by uniform distribution. Assume the distance from the head of the leftmost aisle to the head of the rightmost aisle equals 1.

$$F(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } 0 \leq x \leq 1 \\ 1 & \text{for } x > 1 \end{cases} \tag{4.22}$$

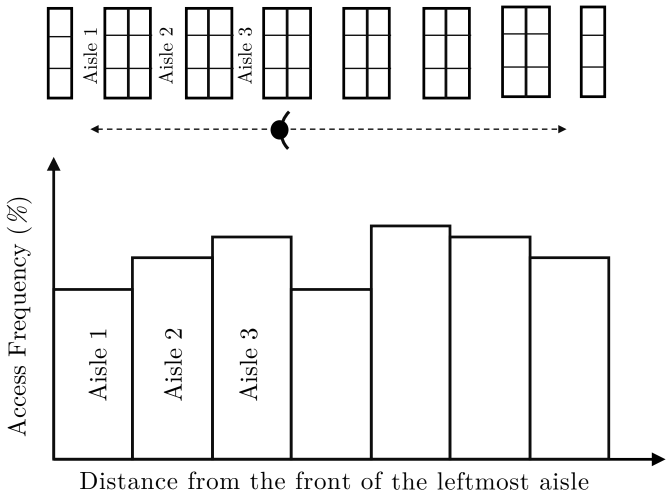


Figure 4.15: Empirical distribution of aisle access frequencies over the distance from the leftmost aisle front

$$f(x) = \begin{cases} 1 & \text{for } 0 \leq x \leq 1, \\ \\ 0 & \text{for } x < 0 \text{ or } x > 1 \end{cases} \tag{4.23}$$

Now imagine an order list containing n items. So we have n independent and identically distributed random variables, i.e, $X_1, X_2, ..., X_n$, with probability distribution function $f(x)$. Relabel these variables such that $X_{(1)} \leq X_{(2)} \leq, ..., \leq X_{(n)}$. So $X_{(k)}$ is the k^{th} smallest X, $k = 1, ..., n$. Using order statistics we can calculate the pdf $f_{X_{(k)}}(x)$.

$$f_{X_{(k)}}(x) = nf(x)\binom{n-1}{k-1}F(x)^{k-1}(1 - F(x))^{n-k} \tag{4.24}$$

Given the uniform distribution assumption for aisle access frequency ($Uniform[0, 1]$), $f_{(X_k)}(x)$ will have a Beta distribution with parameters k and $n-k+1$ ($Beta(k, n-k+1)$). Therefore, the expected value of X_k is:

$$E(X_k) = \frac{k}{n+1} \tag{4.25}$$

Next, we use Equation 4.25 to calculate the expected travel distance of a picker during the setup and picking process under NZ and PZ strategies.

Expected Picker Travel Distance under NZ Strategy

Figure 4.16 presents the travel route of a picker to pick all items for an order. The bold black arrows display the setup distance where the picker travels from her idle location to the front of the first item to pick. The dashed gray arrows show the travel distance for the picker to pick all items. For our analysis, we assume a return routing policy.

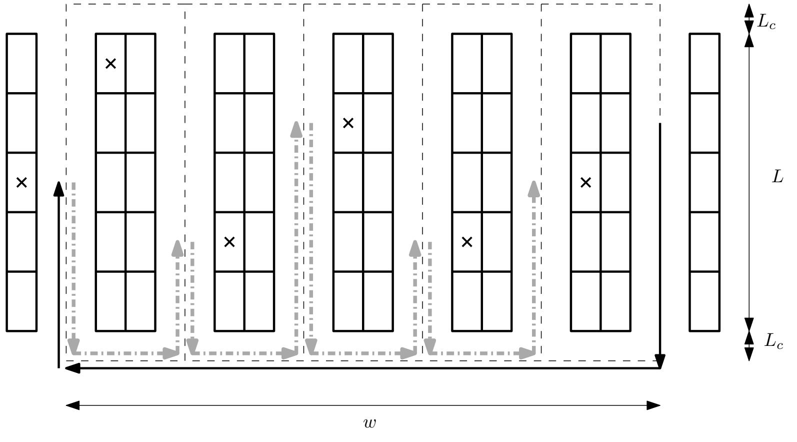


Figure 4.16: Layout of the warehouse

Expected Setup Travel Distance: Setup distance consists of an across-aisle (D_{S_a}), and a within-aisle (D_{S_w}) distance. First, we calculate the across-aisle travel distance. We assume that the picker dwells next to the last picked item in a pick tour. Also, we assume that the pick tour is always from left to right. Let X be the distance from the leftmost aisle to the front of an aisle from which an item needs to be picked. Assuming random storage across aisles, X has a uniform distribution. Let n be the size of the order. Then, the across-aisle distance from the leftmost aisle to the aisle in which the picker dwells is a random variable with pdf $f_{(X_n)}(x)$. Furthermore, the across aisle distance from the leftmost aisle to the aisle form which the first item is to be picked is a random variable with pdf $f_{(X_1)}(x)$. Thus, using Equation 4.25, we can determine the expected across-aisle travel distance of the picker during the setup process.

$$E[D_{S_a}] = w(E[X_n] - E[X_1]) = w\left(\frac{n}{n+1} - \frac{1}{n+1}\right) \tag{4.26}$$

Now we calculate the within-aisle travel distance. Let r be the number of items to pick from each aisle given the order size is n. Let $p(r)$ be the probability of having r items in an aisle when the order size is n. We assume $p(r)$ is the same for every aisle. Let Y be the distance from the head of an aisle to the pick location. Assuming random storage within the aisles, Y has a uniform distribution. Hence, the within-aisle distance from the picker's dwell position to the head of the same aisle is a random variable with pdf $f_{Y_r}(y)$. Furthermore, the within aisle distance from the head of an aisle to the first pick location is a random variable with pdf $f_{Y_1}(y)$. Using Equation 4.25, we can obtain the expected within aisle travel distance of a picker during the setup process.

$$\begin{aligned} E[D_{S_w}] &= \sum_{r=1}^{n} p(r)\left(L_C + L(E[Y_r]) + L_C + L(E[Y_1])\right) \\ &= \sum_{r=1}^{n} p(r)\left(2L_C + L\left(\frac{r}{r+1} + \frac{1}{r+1}\right)\right) \end{aligned} \tag{4.27}$$

Expected Picking Travel Distance: Similarly, the picking travel distance is composed of two components: within-aisle (D_{P_w}) and across-aisle (D_{P_a}) travel distance. It is clear that the expected across-aisle travel distance of the picking operation is the same as the expected across-aisle travel distance of the setup. Therefore, the expression of the expected across-aisle travel distance is the same as shown in Equation 4.26.

Similar to the within-aisle travel distance during the setup operation, let r be the number of items to pick from an aisle. The distance within the first aisle from which an item needs to be picked is to travel from the first pick location to the r^{th} pick location and travel back to the front of the aisle. Hence, the expected travel distance is $L(E[Y_r]) - E[Y_1]) + L_C + L(E[Y_r])$. The distance within the last aisle from which an item needs to be picked is to travel from the front of the aisle to the r^{th} pick location. Therefore, the expected travel distance is $L_C + L(E[Y_r])$. For remaining aisles, the within-aisle travel distance is the distance from the front of the aisle to the r^{th} pick location and travel back to the front of the aisle. Therefore, the expected travel time is $2(L_C + L(E[Y_r]))$. Let N_A be the number of aisles in the warehouse. Equation 4.28 presents the expression for the expected within-aisle travel distance of the picking process.

$$\begin{aligned} E[D_{P_w}] &= \sum_{r=1}^{n} p(r)(L(E[Y_r]) - E[Y_1]) + L_C + L(E[Y_r]) + L_C + L(E[Y_r]) \\ &\quad + 2(N_A - 2)(L_C + L(E[Y_r]))) \\ E[D_{P_w}] &= \sum_{r=1}^{n} p(r)(L((2N_A - 1)E[Y_r] - E[Y_1]) - 2L_C) \\ E[D_{P_w}] &= \sum_{r=1}^{n} p(r)\left(L\left((2N_A - 1)\left(\frac{r}{r+1}\right) - \left(\frac{1}{r+1}\right)\right) - 2L_C\right) \end{aligned} \tag{4.28}$$

Expected Total Picking Travel Distance: The expected total travel distance for the picking process in the NZ strategy equals the sum of setup and pick distances, i.e., $E[D_S]+$

$E[D_P]$. Equation 4.29 presents the expected total travel distance.

$$E[D_{NZ}] = \underbrace{2w\left(\frac{n-1}{n+1}\right)}_{\text{across-aisle distance}} \underbrace{+2N_A\sum_{r=1}^{n} p(r)\left(L_C + L\left(\frac{r}{r+1}\right)\right)}_{\text{within-aisle distance}} \tag{4.29}$$

Expected Picker Travel Distance under PZ Strategy with Two Zones

We are interested in calculating pickers travel distance in one zone when the order size is n. Since we assume a uniform distribution of the pick locations and given the assumption that the probability of skipping any of the zones is zero, each picker will pick on average $n/2$ items in his/her zone. The expected total travel distance is calculated in a similar fashion as in the NZ strategy. The only difference is that the picker covers only her zone. Equation 4.30 presents the expected total travel distance.

$$\begin{aligned} E[D_{PZ}] &= 2\left(\frac{w}{2}\left(\frac{\frac{n}{2}-1}{\frac{n}{2}+1}\right)\right) + 2\left(\frac{N_A}{2}\sum_{r=1}^{\frac{n}{2}} p'(r)\left(L_C + L\left(\frac{r}{r+1}\right)\right)\right) \\ &= \underbrace{w\left(\frac{n-2}{n+2}\right)}_{\text{accross-aisles distance}} \underbrace{+N_A\sum_{r=1}^{\frac{n}{2}} p'(r)\left(L_C + L\left(\frac{r}{r+1}\right)\right)}_{\text{within-aisles distance}} \end{aligned} \tag{4.30}$$

In which, $p'(r)$ is the probability of having r items in an aisle when the order size is n.

Expression for $T_{NZ} - 2T_{PZ}$

We calculate the expected travel time, i.e., T_{NZ} and T_{PZ}, by dividing the expected picker travel distance under each pick strategy by the average speed of the picker (v_p). So, it suffices to obtain the expression for $E[D_{NZ}] - 2E[D_{PZ}]$. Given the assumption in Section 4.6.1, it is easy to see that the expected within-aisles travel distance of the picker under the NZ strategy is double the expected travel distance of a picker in one zone under the PZ strategy. Therefore, to calculate the expression $E[D_{NZ}]-2E[D_{PZ}]$, we only need to consider the across-aisle travel distance. Therefore, using Equations 4.29 and 4.30 we have:

$$E[D_{NZ}] - 2E[D_{PZ}] = 2w\left(\frac{2n}{(n+1)(n+2)}\right)$$

And finally we have:

$$T_{NZ} - 2T_{PZ} = \frac{2w}{v_p}\left(\frac{2n}{(n+1)(n+2)}\right) \tag{4.31}$$

4.10.4 Order Data Description

We evaluate our model with order data from a warehouse of a logistics company in the Netherlands. The data file includes 88,440 orders and a total of 1.8 million order lines. Note that the number of order lines is the number of unique items present in each order. However, the quantity of each order line in an order can be more than one. Figure 4.17 shows the frequency of the top 100 items ordered by the customers. The total number of unique items present in the warehouse is 6058. The order frequency of the items has a typical skewed distribution, where the top 30 percent of the items make up for 82.68 percent of the demand.

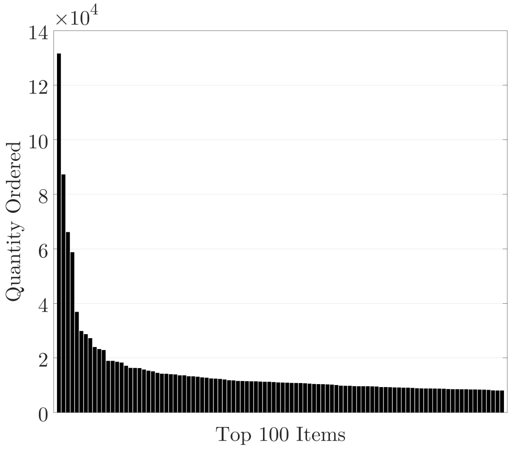


Figure 4.17: Quantity ordered of the top 100 items from a warehouse of a logistics company in the Netherlands

In our analysis, we are interested to see the distribution of order sizes. Table 4.10 presents the frequency of lines per order. We focus on orders with 80 or fewer order lines in our analysis for two reasons. First, they constitute about 81.29 percent of all orders and they maintain the integrity of the order size distribution. Second, since we assume every order is fulfilled within a single pick tour, orders of 80 or more are fulfilled in multiple pick routes due to the capacity constraint of the order bins on top of the robots. Figure 4.18 presents the distribution of the orders of 80 or fewer. From the order frequency of the items as well as the distribution of the frequency of different order sizes, it is evident that the data is from a multi-channel warehouse.

In the numerical analysis, we distinguish small and large-sized orders and their frequency of occurrence based on this data. Small-sized orders are those with five or fewer (possibly for e-commerce channel) and large-sized are between five and 80 orders (probably for store channel). Using this definition, 67.05 percent of the orders are small-size (typically e-

Table 4.10: Frequency of lines per order

Number of Lines/Order	Frequency	Percentage
1	28238	31.93%
2	8193	9.26%
3	5662	6.40%
4	3383	3.83%
5	2136	2.42%
6	1775	2.01%
7	1244	1.41%
8	1157	1.31
.	.	.
.	.	.
>80	17438	19.71%

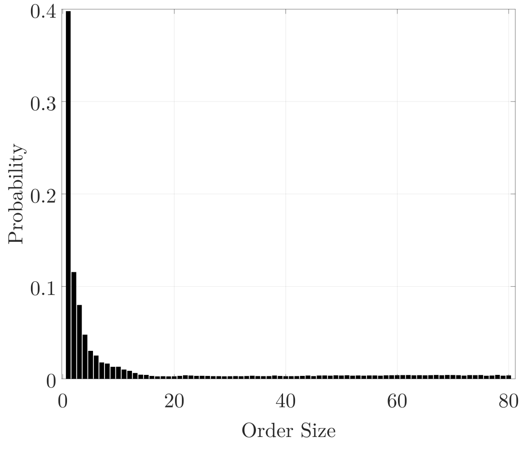


Figure 4.18: Distribution of orders with 80 or fewer order lines

commerce) orders, and 32.95 percent of the orders are large-sized (typically store) orders. Table 4.11 shows the descriptive statistics of the small- and large-sized orders based on the distribution presented in Figure 4.18.

Table 4.11: Descriptive statistics of small and large size orders

	Count	Mean	Std	Min	25%	50%	75%	Max
Small-sized orders	45476	1.65	0.96	1	1	1	4	20
Large-sized orders	23390	35.39	24.71	6	10	32	58	80

4.10.5 Instances for Validation of the Solution Methods for Network 1 and Network 2

Note: half-widths are based on 95 percent confidence level.

Instances for Network 1:

Table 4.12: Throughput

# Picker	Depot Time	# Robots	Analytical	Simulation	Half-width	Error
2	10	2	0.043435419	0.043461873	0.000234841	0.06%
2	10	4	0.060842014	0.060104761	0.000332316	1.23%
2	10	6	0.065847933	0.066685584	0.0003392	1.26%
2	10	8	0.067596033	0.069788927	0.000375433	3.14%
2	10	10	0.068278839	0.07169752	0.000434299	4.77%
2	15	2	0.037505866	0.037600462	0.000196895	0.25%
2	15	4	0.052204324	0.051596125	0.000343775	1.18%
2	15	6	0.057467486	0.057350163	0.000321609	0.20%
2	15	8	0.060053046	0.06029394	0.000298433	0.40%
2	15	10	0.061584809	0.062268905	0.000339353	1.10%
4	10	2	0.043587629	0.043351034	0.000233648	0.55%
4	10	4	0.075395847	0.075266125	0.000268128	0.17%
4	10	6	0.090182603	0.089045395	0.000433974	1.28%
4	10	8	0.095719061	0.095174939	0.000439551	0.57%
4	10	10	0.098013461	0.097796046	0.000475876	0.22%
4	15	2	0.037579126	0.037520398	0.000223901	0.16%
4	15	4	0.058587521	0.058482869	0.00030621	0.18%
4	15	6	0.064981149	0.064903467	0.000453164	0.12%
4	15	8	0.06632606	0.065878831	0.000516718	0.68%
4	15	10	0.066597549	0.066524897	0.000426527	0.11%

Table 4.13: Utilization OPS

# Picker	Depot Time	# Robots	Analytical	Simulation	Half-width	Error
2	10	2	0.434354192	0.4357	0	0.30%
2	10	4	0.608420138	0.6038	0	0.76%
2	10	6	0.658479327	0.6651	0	0.99%
2	10	8	0.675960335	0.7010	0.01	3.57%
2	10	10	0.682788385	0.7206	0.01	5.24%
2	15	2	0.562587988	0.5599	0	0.49%
2	15	4	0.783064859	0.7771	0	0.76%
2	15	6	0.862012288	0.8641	0	0.24%
2	15	8	0.900795685	0.9049	0	0.46%
2	15	10	0.923772141	0.9321	0	0.89%
4	10	2	0.435876287	0.4324	0	0.80%
4	10	4	0.753958472	0.7552	0	0.17%
4	10	6	0.901826029	0.8948	0	0.79%
4	10	8	0.957190613	0.9503	0	0.72%

Continued on next page

Table 4.13 – *Continued from previous page*

# Picker	Depot Time	# Robots	Analytical	Simulation	Half-width	Error
4	10	10	0.980134609	0.9777	0	0.25%
4	15	2	0.563686893	0.5607	0	0.53%
4	15	4	0.878812819	0.8786	0	0.03%
4	15	6	0.974717237	0.9741	0	0.07%
4	15	8	0.994890905	0.9943	0	0.06%
4	15	10	0.998963238	0.9990	0	0.00%

Table 4.14: Mean number of Jobs OPS

# Picker	Depot Time	# Robots	Analytical	Simulation	Half-width	Error
2	10	2	0.533864889	0.5364	0	0.48%
2	10	4	1.083564127	1.0622	0.01	2.01%
2	10	6	1.465066493	1.4663	0.02	0.09%
2	10	8	1.699566503	1.8305	0.03	7.15%
2	10	10	1.833285383	2.1274	0.05	13.82%
2	15	2	0.736145448	0.7304	0	0.79%
2	15	4	1.703062776	1.6639	0.01	2.35%
2	15	6	2.732688808	2.7175	0.03	0.56%
2	15	8	3.798659367	3.8254	0.04	0.70%
2	15	10	4.893331193	5.0379	0.09	2.87%
4	10	2	0.5358697	0.5310	0	0.91%
4	10	4	1.463354979	1.4627	0.01	0.05%
4	10	6	2.756889198	2.6860	0.02	2.64%
4	10	8	4.28830353	4.1000	0.03	4.59%
4	10	10	5.98561285	5.7818		3.53%
4	15	2	0.737698344	0.7343	0	0.46%
4	15	4	2.030586107	2.0337	0.01	0.15%
4	15	6	3.752321797	3.7255	0.02	0.72%
4	15	8	5.661683535	5.6217	0.03	0.71%
4	15	10	7.635856986	7.5765	0.02	0.78%

Table 4.15: Utilization MTPS

# Picker	Depot Time	# Robots	Analytical	Simulation	Half-width	Error
2	10	2	0.631726909	0.6311	0	0.10%
2	10	4	0.884889291	0.8588	0	3.04%
2	10	6	0.957695625	0.9357	0	2.35%
2	10	8	0.983120091	0.9657	0	1.81%
2	10	10	0.993050842	0.9818	0	1.15%
2	15	2	0.545487188	0.5487	0	0.58%
2	15	4	0.759262298	0.7420	0	2.33%
2	15	6	0.835809987	0.8122	0.005	2.90%
2	15	8	0.873414499	0.8499	0	2.77%
2	15	10	0.895692547	0.8712	0.01	2.81%
4	10	2	0.316970326	0.3189	0.0025	0.62%
4	10	4	0.548280486	0.5490	0	0.13%

Continued on next page

Table 4.15 – *Continued from previous page*

# Picker	Depot Time	# Robots	Analytical	Simulation	Half-width	Error
4	10	6	0.655810143	0.6463	0	1.48%
4	10	8	0.696071406	0.6882	0.005	1.15%
4	10	10	0.712756338	0.7012	0.0025	1.65%
4	15	2	0.273276345	0.2740	0.005	0.27%
4	15	4	0.426049919	0.4256	0.0025	0.10%
4	15	6	0.472544541	0.4698	0.0025	0.59%
4	15	8	0.482324769	0.4790	0.0075	0.69%
4	15	10	0.484299043	0.4841	0.005	0.05%

Table 4.16: Mean Number of Jobs at MTPS

# Picker	Depot Time	# Robots	Analytical	Simulation	Half-width	Error
2	10	2	1.098132865	1.26970284	0	13.51%
2	10	4	2.400957996	2.63320875	0.03	8.82%
2	10	6	3.977043483	4.15699383	0.03	4.33%
2	10	8	5.727732863	5.75083692	0.07	0.40%
2	10	10	7.588228985	7.42701432	0.11	2.17%
2	15	2	0.946089855	1.10184400	0	14.14%
2	15	4	1.854641309	2.07930945	0.01	10.80%
2	15	6	2.780423665	2.97836888	0.05	6.65%
2	15	8	3.692547209	3.83997914	0.06	3.84%
2	15	10	4.584897668	4.60722739	0.11	0.48%
4	10	2	1.094838475	1.2758	0.01	14.18%
4	10	4	1.897861245	2.2005	0	13.75%
4	10	6	2.479047717	2.8964	0	14.41%
4	10	8	2.900726295	3.4337	0.07	15.52%
4	10	10	3.183977904	3.7346	0.6	14.74%
4	15	2	0.943916267	1.0960	0.02	13.88%
4	15	4	1.473036978	1.7038	0.01	13.55%
4	15	6	1.697131916	1.9756	0.01	14.10%
4	15	8	1.776375552	2.0743	0.03	14.36%
4	15	10	1.799901939	2.1147	0.02	14.88%

Instances for Network 2:

Table 4.17: Throughput

# Picker	Depot Time	# Robots	Analytical	Simulation	Half-width	Error
2	10	2	0.031412167	0.0318	0.0002	1.27%
2	10	4	0.042812218	0.0446	0.0004	3.98%
2	10	6	0.047408229	0.0504	0.0004	6.01%
2	10	8	0.049417123	0.0532	0.0004	7.05%
2	10	10	0.050326087	0.0541	0.0005	6.94%
2	15	2	0.028672131	0.0292	0.0002	1.92%
2	15	4	0.039682953	0.0413	0.0003	3.95%

Continued on next page

Table 4.17 – *Continued from previous page*

# Picker	Depot Time	# Robots	Analytical	Simulation	Half-width	Error
2	15	6	0.044752864	0.0471	0.0003	4.92%
2	15	8	0.047413877	0.0505	0.0003	6.10%
2	15	10	0.04891348	0.0521	0.0004	6.17%
4	10	2	0.039084619	0.0378	0.0002	3.38%
4	10	4	0.063931764	0.0634	0.0003	0.88%
4	10	6	0.076775975	0.0771	0.0004	0.43%
4	10	8	0.083702789	0.0845	0.0004	0.98%
4	10	10	0.087799671	0.0894	0.0004	1.83%
4	15	2	0.034335023	0.0333	0.0002	3.07%
4	15	4	0.053104148	0.0526	0.0003	0.97%
4	15	6	0.061033253	0.0611	0.0003	0.11%
4	15	8	0.064296211	0.0644	0.0004	0.09%
4	15	10	0.065661647	0.0661	0.0004	0.65%

Table 4.18: Utilization OPS

# Picker	Depot Time	# Robots	Analytical	Simulation	Half-width	Error
2	10	2	0.314121668	0.3190	0	1.52%
2	10	4	0.428122185	0.4462	0	4.06%
2	10	6	0.474082286	0.5046	0.01	6.06%
2	10	8	0.494171229	0.5325	0.01	7.21%
2	10	10	0.503260867	0.5404	0.01	6.87%
2	15	2	0.430081971	0.4351	0.01	1.15%
2	15	4	0.595244295	0.6207	0.01	4.10%
2	15	6	0.671292956	0.7028	0.01	4.48%
2	15	8	0.711208154	0.7595	0.01	6.35%
2	15	10	0.733702206	0.7828	0.01	6.27%
4	10	2	0.390846193	0.3822	0	2.26%
4	10	4	0.639317643	0.6342	0	0.80%
4	10	6	0.767759753	0.7709	0	0.40%
4	10	8	0.837027886	0.8470	0	1.17%
4	10	10	0.877996714	0.8968	0	2.10%
4	15	2	0.515025347	0.5020	0	2.60%
4	15	4	0.796562214	0.7909	0	0.71%
4	15	6	0.915498798	0.9172	0	0.18%
4	15	8	0.964443171	0.9702	0	0.59%
4	15	10	0.984924704	0.9873	0	0.24%

Table 4.19: Mean Number of Jobs at OPS

# Picker	Depot Time	# Robots	Analytical	Simulation	Half-width	Error
2	10	2	0.375998407	0.3758	0	0.05%
2	10	4	0.654646421	0.6738	0.01	2.84%
2	10	6	0.831732324	0.8841	0.02	5.92%
2	10	8	0.934197642	1.0293	0.03	9.24%
2	10	10	0.989570712	1.0812	0.03	8.47%

Continued on next page

Table 4.19 – *Continued from previous page*

# Picker	Depot Time	# Robots	Analytical	Simulation	Half-width	Error
2	15	2	0.375998407	0.3758	0	0.05%
2	15	4	0.654646421	0.6738	0.01	2.84%
2	15	6	0.831732324	0.8841	0.02	5.92%
2	15	8	0.934197642	1.0293	0.03	9.24%
2	15	10	0.989570712	1.0812	0.03	8.47%
4	10	2	0.471685898	0.4573	0	3.15%
4	10	4	1.142351655	1.1011	0.01	3.74%
4	10	6	1.942160168	1.8702	0.02	3.85%
4	10	8	2.825078693	2.7714	0.04	1.94%
4	10	10	3.766076075	3.8081	0.05	1.10%
4	15	2	0.659835601	0.6368	0	3.61%
4	15	4	1.7009287	1.6445	0.01	3.43%
4	15	6	3.064515331	2.9801	0.03	2.83%
4	15	8	4.66985743	4.6385	0.03	0.68%
4	15	10	6.440091028	6.3672	0.05	1.14%

Table 4.20: Utilization MTPS1

# Picker	Depot Time	# Robots	Analytical	Simulation	Half-width	Error
2	10	2	0.413891421	0.4188	0.01	1.18%
2	10	4	0.564100213	0.5793	0.01	2.63%
2	10	6	0.624657931	0.6434	0.01	2.92%
2	10	8	0.651127424	0.6767	0.01	3.78%
2	10	10	0.663104067	0.6878	0.01	3.59%
2	15	2	0.377788304	0.3837	0.01	1.55%
2	15	4	0.522868541	0.5347	0.01	2.20%
2	15	6	0.589670445	0.6117	0.01	3.60%
2	15	8	0.624732355	0.6490	0.01	3.75%
2	15	10	0.644491354	0.6689	0.01	3.65%
4	10	2	0.257492403	0.2492	0.005	3.33%
4	10	4	0.421187258	0.4170	0.005	1.00%
4	10	6	0.505805884	0.5041	0.01	0.35%
4	10	8	0.551440249	0.5512	0.01	0.05%
4	10	10	0.57843082	0.5819	0.01	0.59%
4	15	2	0.226201708	0.2218	0	2.00%
4	15	4	0.349854107	0.3526	0	0.79%
4	15	6	0.402091649	0.3999	0.01	0.56%
4	15	8	0.423588263	0.4203	0.01	0.77%
4	15	10	0.432583855	0.4346	0.01	0.46%

Table 4.21: Mean Number of Jobs at MTPS1

# Picker	Depot Time	# Robots	Analytical	Simulation	Half-width	Error
2	10	2	0.497349723	0.51659733	0	3.73%
2	10	4	0.966189842	0.97759415	0.03	1.17%
2	10	6	1.33851603	1.32930092	0.04	0.69%

Continued on next page

Table 4.21 – *Continued from previous page*

# Picker	Depot Time	# Robots	Analytical	Simulation	Half-width	Error
2	10	8	1.603743243	1.62988812	0.07	1.60%
2	10	10	1.77765625	1.82813087	0.08	2.76%
2	15	2	0.445178218	0.46343805	0	3.94%
2	15	4	0.849697297	0.85962252	0.03	1.15%
2	15	6	1.178468079	1.20152820	0.03	1.92%
2	15	8	1.431429387	1.45892169	0.04	1.88%
2	15	10	1.618516818	1.62134508	0.07	0.17%
4	10	2	0.448550622	0.49890492	0.01	10.09%
4	10	4	0.826384094	0.90517755	0	8.70%
4	10	6	1.12582414	1.21962299	0.02	7.69%
4	10	8	1.353017474	1.45811750	0.04	7.21%
4	10	10	1.522199341	1.64227987	0.05	7.31%
4	15	2	0.393051513	0.44380962	0	11.44%
4	15	4	0.667231941	0.75411174	0	11.52%
4	15	6	0.830974105	0.91253753	0.02	8.94%
4	15	8	0.917715739	0.99764595	0.03	8.01%
4	15	10	0.960033226	1.05863797	0.03	9.31%

Table 4.22: Utilization MTPS2

# Picker	Depot Time	# Robots	Analytical	Simulation	Half-width	Error
2	10	2	0.61481542	0.6280	0.01	2.11%
2	10	4	0.837943217	0.8476	0	1.14%
2	10	6	0.92789874	0.9352	0	0.78%
2	10	8	0.967217873	0.9711	0	0.40%
2	10	10	0.985008589	0.9861	0	0.11%
2	15	2	0.561186008	0.5752	0.01	2.44%
2	15	4	0.776695589	0.7843	0.01	0.97%
2	15	6	0.875926544	0.8782	0	0.26%
2	15	8	0.928009292	0.9252	0	0.30%
2	15	10	0.957360317	0.9548	0	0.27%
4	10	2	0.382492344	0.3696	0	3.49%
4	10	4	0.625653027	0.6159	0.005	1.59%
4	10	6	0.751349848	0.7376	0.01	1.87%
4	10	8	0.819137461	0.8037	0.005	1.93%
4	10	10	0.859230632	0.8397	0	2.33%
4	15	2	0.33601155	0.3260	0	3.07%
4	15	4	0.519691128	0.5096	0	1.98%
4	15	6	0.597287437	0.5935	0.01	0.64%
4	15	8	0.629219603	0.6167	0.005	2.03%
4	15	10	0.642582113	0.6355	0.01	1.12%

Table 4.23: Mean Number of Jobs at MTPS2

# Picker	Depot Time	# Robots	Analytical	Simulation	Half-width	Error
2	10	2	0.794691233	0.85040788	0	6.55%

Continued on next page

Table 4.23 – *Continued from previous page*

# Picker	Depot Time	# Robots	Analytical	Simulation	Half-width	Error
2	10	4	1.926728493	1.94839741	0.02	1.11%
2	10	6	3.328746227	3.30024658	0.04	0.86%
2	10	8	4.939823902	4.82389198	0.08	2.40%
2	10	10	6.700931986	6.55574595	0.09	2.21%
2	15	2	0.706051075	0.75897350	0	6.97%
2	15	4	1.647998756	1.65375735	0.02	0.35%
2	15	6	2.782726861	2.70299208	0.03	2.95%
2	15	8	4.089454884	3.85878273	0.06	5.98%
2	15	10	5.548947896	5.24628373	0.1	5.77%
4	10	2	0.666721132	0.74020452	0.01	9.93%
4	10	4	1.355639759	1.47877653	0.02	8.33%
4	10	6	2.120654862	2.24902356	0.05	5.71%
4	10	8	2.937341133	3.02173446	0.05	2.79%
4	10	10	3.783866437	3.74357883	0.06	1.08%
4	15	2	0.584263794	0.65245432	0	10.45%
4	15	4	1.070640038	1.17133665	0	8.60%
4	15	6	1.459517247	1.59383421	0.05	8.43%
4	15	8	1.732950898	1.81758463	0.04	4.66%
4	15	10	1.905970027	2.00684264	0.06	5.03%

5 Conclusions and Future Outlook

In this dissertation, we explore new automated and robotic order picking systems designed to boost warehouse productivity. In the introduction chapter, we introduce different types of warehouses that have emerged as a result of multi-channel retail operations. We then explore how different channels require different automated and robotic order picking solutions. In the subsequent chapters, we first explore recent trends in warehouse automation and robotics. We then choose two robotic solutions for further investigation. Traditional automated systems are known for throughput inflexibility (Roodbergen & Vis, 2009). For example, in an Automated Storage and Retrieval System (AS/RS) with aisle-captive cranes, the throughput capacity is bounded by the number of cranes in the system and cannot be expanded. However, most robotic solutions provide flexible throughput capacity since the robots can be dynamically added or retrieved from the system to adjust the throughput to the desired level. The dynamic models that we develop in this dissertation can be used to test different design trade-offs and operational strategies and to support real-time decision-making. This concluding chapter summarizes the key results and insights from our research and discusses several directions for future studies.

5.1 Conclusions

In Chapter 2, we present an overview of the recent trends in automated warehousing, particularly the use of robotic technologies for order fulfilment. The volatile demand in an e-commerce environment requires a high level of throughput flexibility. Although manual picking provides greater throughput flexibility compared to robotic picking, finding high-quality labor has become challenging in many parts of the world. Furthermore, unexpected major disruptions such as Brexit and Covid-19 pandemic have made it even harder to find and operate with manual labors. Robotic solutions provide savings in floor space, labor, and operational costs while providing enough scalability and throughput flexibility to meet demand fluctuations. Therefore, they are an appealing alternative for manual picking.

Automation of the storage and retrieval process requires a long-term investment and, therefore, a long-term vision. We distinguish two decision-making levels in warehouse planning and design: long-term (tactical) and short-term (operational). The long-term decisions revolve around hardware design selection and optimization of the system. The prime objectives at this stage are to maximize the throughput and the storage capacity of the system against the cost. The short-term decisions revolve around operational planning and control. The prime objectives at this level are to minimize order lead time, waiting time, response time, and resource idleness. Academic studies have developed several models to optimize the performance of the various automated systems, focusing on short- and long-term decisions. We present these different modeling techniques as well as their corresponding solution approaches to evaluate the performance of the automated systems. We then provide a decision-making framework that includes different objectives, the corresponding decision variables, and the suitable modeling approaches to address them.

Next, we describe well-established automation technologies, including AS/R systems with cranes or automated forklifts, carousels, vertical lift modules, automated dispensing systems, aisle-based shuttle systems, grid-based shuttle systems, and robotic mobile fulfillment systems (RMF). We review the literature related to the various design and control problems in these systems, such as optimizing the shape of the system, finding an optimal storage assignment, and investigating the impact of vehicle dwell point policies on the system performance. Next, we identify promising new emerging technologies that, to date, have hardly received any research attention, including vertical and diagonal robotic storage and retrieval systems, robot-based compact storage and retrieval systems (RCSR), gridsort, and pick-support autonomous mobile robots (PS-AMRs). We conclude this chapter by summarizing the unaddressed research questions in established systems and pose research questions for emerging technologies.

In Chapter 3, we investigate the vertical storage and retrieval system, an emerging robotic technology for e-commerce warehouses. The shuttle- or autonomous vehicle-based storage and retrieval systems (AVS/R) use a combination of autonomous shuttles and lifts to perform the order fulfillment. Shuttles move autonomously in the horizontal directions and are transported in the vertical direction using lifts. Therefore, we categorize them as horizontal systems. The system throughput in the horizontal systems is constrained by the number of lifts in this system. The vertical systems have been developed to address this problem. In these systems, a single robot can independently roam throughout the storage rack to transport items between storage locations and workstations. Therefore, the vertical system has the edge over its horizontal counterpart when it comes to throughput flexibility. In a given vertical system, the desired throughput level can only be obtained by choosing the correct number of robots. However, in a given horizontal system, the number of shuttles, as well as the number of lifts, need to be adjusted to achieve a certain throughput rate. The prime contributions of this study are as follows:

1. We are the first to investigate these vertical robotic-based storage systems. They are fundamentally different from the horizontal systems since robots move horizontally as well as vertically to perform order transactions. Therefore, new models are required to capture their performance.

2. Because a large number of robots in the system may lead to blocking and delays we propose different block prevention policies. We present a jump-over (Van Dijk, 1988) approximation method to analyze a block prevention policy. This study is the first to use this technique in a robotic warehousing context.

3. We are the first to investigate the optimal system size that maximizes system throughput for the new generation robotic-based vertical system. We also compare the effect of two blocking policies on the system throughput capacity. Furthermore, we are the first to compare cost as well as system throughput performance of vertical systems and the horizontal systems.

We first model the system, assuming there is no congestion nor blocking in the system. We develop a closed queuing network and estimate the system throughput using the AMVA method (approximate mean value analysis). We then develop an optimization model to find the optimal height to width ratio of each aisle that maximizes the system throughput. We then relax the no congestion assumption and investigate two protocols for the blocked robots, Wait-on-Spot (WOS) and Recirculation (REC). In the WOS policy, the robots wait on top of the occupied rack section until it is unoccupied. In the REC policy, a blocked robot keeps recirculating in the outer path around the occupied rack section until the designated rack section is empty. We develop two closed queuing networks corresponding to each policy to estimate the performance of the system. The system throughput in the WOS network is estimated by using an approximate method proposed by Akyildiz (1988). We approximate the REC network by another one with the jump-over blocking protocol (Van Dijk, 1988) and use an iterative algorithm based on the MVA method to estimate the system throughput. Finally, we compare the cost-performance of the vertical system, the horizontal system with a discrete lift (horizontal-d), and the horizontal system with a continuous lift (horizontal-c). The horizontal system is modeled as a closed queuing network with an unlimited buffer, and the system throughput is estimated using AMVA. We derive the following results and managerial insights:

- *Optimal Rack Layout Configuration:* In terms of throughput performance, we show that it is better to have a system with equal length and hight. Using a probabilistic travel time approach instead of a queuing network also leads to similar results. Therefore, it can be concluded that the waiting times in the system do not affect the optimal shape of the system.

- *Block Prevention Policies Comparison:* When comparing the results of the two policies, the WOS has a slight advantage over the REC policy when the number of robots in the system is small. However, increasing the number of robots results in a sharp decrease in system throughput. The REC has a lower system throughput, especially when the waiting time is less than the recirculation time, and the number of robots is small. However, the REC policy can achieve a higher system throughput without worrying about the blocking delays when the number of robots in the system increases. Therefore, neither of the two blocking policy dominates the other one. Hence, a company should choose the most suitable depending on the number of robots in the system and the desired throughput level.

- *Cost-Performance Comparison:* When there is one load/unload (L/U) point in the system, the vertical system outperforms the horizontal-d system in both operating costs and system throughput. However, compared with the horizontal-c system, its performance depends on the number of storage locations in the system. When there are two L/U points in the system, the vertical system always has a lower operating cost than horizontal systems. The reason behind this outcome is that the horizontal system uses two material handling resources (lift as well as shuttles), whereas the vertical system only uses the rack climbing robots. The cost difference is even more notable when the systems have two L/U points. Because introducing an additional L/U point in the horizontal systems requires installing an additional lift. Whereas, in a vertical system, an extra L/U point can be added without requiring the installation of any additional material handling system. Therefore, the operating cost to achieve a certain throughput capacity in the vertical system is significantly lower.

In Chapter 4, we study systems in which a human picker collaborates with robots to fulfill orders. Such solutions are necessary, as we are still several years from observing a fully automated robotized system. Robot intelligence is still far from matching the full range of human picker capabilities, especially handling odd-shaped items appropriately and at high speed. Hence, human-robot collaborative systems have been developed. In these systems, PS-AMRs assist human pickers in fulfilling orders. In human-robot collaborative systems, different operational policies can be embedded in the robot control software, which allows to dynamically adjust the system's behavior to improve pick performance. One of the decisions that can significantly improve the pick performance is zoning. In this chapter, we focus on two pick strategies: the No Zoning (NZ) strategy and the Progressive Zoning (PZ) strategy. In the PZ strategy, the warehouse is divided into multiple storage zones, with one or multiple pickers assigned to each zone. Pickers only pick from their dedicated zones while robots can move to any storage zone. There are operational trade-offs involved in the selection of the two pick strategies. While the robots can travel to any pick location in both strategies, the pickers' movement is restricted, depending on the pick strategy. In the PZ strategy, the pickers travel time is reduced because her movement is limited to the zone. In this strategy, a partially-filled order is picked in the next zone by a different picker. However, due to demand variability among the zones, the robot may have to wait in a zone for an available picker. In the NZ strategy, a robot's waiting time to access an available picker is reduced since pickers may access any pick location within the warehouse. However, the pickers travel time per order can increases because she may have to visit locations throughout the warehouse.

We then investigate the effect of dynamic zoning decisions (dynamically switching between NZ and PZ strategy) on the performance of the collaborative picking system. The prime contributions of this study are as follows:

1. We are the first to develop a stochastic model for a human-robot collaborative picking system.
2. We develop a novel closed-queuing model using the technique of two-phase queuing servers (Van Doremalen, 1986) to capture the parallel movement of the picker and the robot.
3. We are the first to investigate the effect of dynamic zoning on system performance.

We develop queuing network models to estimate the performance of each picking strategy. A two-phase queuing process is used to accurately capture the parallel movement of the resources in the system, i.e., robots and human pickers. The results show that the average order size, i.e., the number of items on the pick list, affects the best choice of the zoning strategy. However, in a warehouse with different order sizes such as omni-channel warehouses, a fixed picking strategy might not be the best option. Therefore, we develop an MDP model to obtain a DS (Dynamic Switching) policy based on the number of small- and large-sized orders in the system. We derive the following results and managerial insights:

- *Fixed Policies:* The results show that the PZ strategy has a higher throughput performance than the NZ strategy when order sizes are small because the expected travel distance from one zone to another is relatively long. Therefore, by using the PZ strategy and letting the robots travel this distance, shorter throughput time, and hence a higher throughput can be achieved. In contrast, when order sizes are large, the NZ outperforms the PZ strategy since it reduces the robot's waiting time to access an available picker. Therefore, the PZ strategy is suitable in e-commerce warehouses with predominantly small-sized customer orders, and the NZ strategy is suitable for store replenishment warehouses with predominantly large-sized orders.
- *Dynamic Policies:* The results show when the number of operating robots per picker in a warehouse is large (about two), the DS policy does not generate substantial operational cost savings. However, with a small number of robots per picker (about 1.3), the DS policy can reduce operational costs by up to 7 percent when 55 percent of the incoming orders are small. This observation is particularly important for companies that want to automate their picking operation gradually. They can start with a small number of robots and operate under the DS policy and switch to a fixed policy when the robot fleet has increased to a certain level.

5.2 Future Outlook

This dissertation explored new automated and robotic order fulfillment technologies. The results reveal valuable insights and point to several exciting and important aspects that require further investigation. We provide directions for future research concerning the systems that have been studied in this dissertation, as well as several warehousing aspects that need further academic studies.

Vertical Storage and Retrieval System

One of the vertical system's unique features is the flexibility of the robots to choose different routing trajectories while processing storage and retrieval transactions. Therefore, it would be interesting for future studies to find the robots' routing trajectories that improve system performance.

Due to the inherent differences between the vertical and horizontal systems, the optimal storage policies that have been developed for horizontal systems (e.g., Kuo et al. (2008), Roy et al. (2012)) might not be suitable for the vertical systems. Therefore, studies are needed to examine the effect of different storage policies on the vertical system's performance.

Moreover, in the presence of multi-line orders, the sequence in which the items are retrieved from the system can potentially improve the picking operation's efficiency. Therefore, it would be interesting to investigate the effect of transaction sequencing policies on system throughput.

Pick-Support AMRs

Research on collaborative AMRs is still in its infancy, and there are several potential directions for future studies. In Chapter 4, we study the effect of zoning the storage area with dedicated pickers. However, we limit ourselves to a fixed number of zones and fixed zone boundaries. When a picker is assigned to pick from the first three aisles, as long as the picking strategy has not changed, she continues to pick only from those three aisles. However, the PS-AMR system can operate with dynamic zone boundaries. Meaning, depending on the workload, each zone's size can be increased or decreases dynamically, leading to a higher pick performance. Therefore, it would be interesting for future research to examine the effect of dynamic zone boundaries on the pick performance.

Furthermore, in our model, we ignore the transition time to switch between pick strategies and assume once a picking strategy changes, the system immediately starts operating under the other strategy. However, in reality, some transition time is required to switch from one strategy to another. For example, if the strategy changes from NZ to PZ, it takes some time for the pickers to move to their designated zones before the picking can resume. Therefore, and interesting extension of our model would be to include the transition time of switching among different strategies.

Moreover, PS-AMRs operate with batteries with limited capacity. Therefore, a potential extension can be to investigate a picking strategy considering robots' battery capacity and charging requirements.

Human-Machine Interaction

Most automated systems described in Chapter 2 retrieve unit loads and bring them to an order pick station. Even with new technologies, like RCSR systems, puzzle-based systems, or RMF systems, piece picking at the station is done manually (De Koster & Yu, 2008; Füßler & Boysen, 2017a, 2019). Furthermore, a human picker is also involved for piece picking in PS-AMR systems. However, new technologies, like deep learning and rapid image processing, are developing that make automated product recognition, selection of the appropriate gripper, and rapid automated picking with robots possible. The Pick-it-Easy robot developed by Knapp is an example of an industrial robot developed for automated picking at a pick station, and the Fetch mobile manipulator developed by Fetch Robotics is an example of a picking AMR (see Figure 5.1).

(a) Fetch mobile manipulator

(b) Pick-it-Easy robot

Figure 5.1: Robot picking (Source: Fetch Robotics and Knapp)

For the coming decade or so, robot picking stations or robot picking AMRs alone cannot do the job cost-efficiently and at sufficient speed, which means humans have to collaborate with robots. This collaboration calls for further study of human-machine interaction. Therefore, it is important to study how the two should cooperate to maximize performance. However, the interaction between man and machine has received little attention in operations management literature. Researchers may focus on addressing the following questions:

- What types of human jobs should remain to maximize joint performance during tasks in cooperation with machines?
- How can we minimize the discomfort of order pickers (see also Larco et al. (2017))?

Or researchers may focus on addressing more behavioral questions:

- How do we incentivize people, or what type of personality should a person have to maximize joint work performance?

A recent study showed that the organization of the pick process, work incentives, and personalities of the pickers strongly interact and can have a major effect on picking performance (De Vries et al., 2016).

Modular System Analysis

To date, academic research on system design is reactive; that is, researchers evaluate designs after implementation. Is it possible to develop a modular system analysis architecture to analyze different design elements and propose new system configurations? Solving this problem presents significant challenges because of the large design space, interaction between design elements, and the diverse needs of warehouse designers. So the question is, how can we develop a modeling framework which assists in analyzing design options and provides quick insights for developing new system designs? By using a modular system analysis, the feasible design space can be shrunk. Then design optimization would be much easier in the reduced feasible space.

New Methods and Modeling Approaches

In addition to the methods discussed in this dissertation, new techniques might need to be developed, or other existing tools can be used that have not been used in a warehousing context to evaluate the automated systems' performance. For instance, data-driven techniques such as data envelopment analysis can be used to benchmark automated systems. New modeling methods need to be developed, for example, for performance evaluation with non-stationary transaction arrivals (see Dhingra et al. (2018)). Many of the recent robotic solutions (e.g., vertical AVS/R, PS-AMRs, RMF systems) are much more flexible in capacity, portability, and extension options than conventional automated storage and retrieval systems. In these systems, the number of robots can be adjusted, and workstations can be opened or closed, depending on the needed capacity. These new capabilities make it possible to dynamically optimize warehouse decisions using real-time data, resulting in higher performance. The questions that justify attention include: when to scale up or down retrieval capacity, when to open or close pick stations, when to reallocate tasks to workers, and when to modify the system layout or storage strategies (e.g., in RMF systems (see Lamballais et al. (2017a)) or PS-AMRs). These decisions can be taken at any point in time and some can be executed rapidly. New data-driven techniques, such as deep learning, may find an application in answering these questions.

Comprehensive Evaluation of Automated Systems

With the rapid introduction of new automation technologies, distribution center managers are confronted with multiple technology options. Therefore, it is crucial to create a framework to help distribution center managers in evaluating the performance of different automated systems. In other words, how can we develop a framework for the comprehensive evaluation of technologies against cost and time? The only (published) paper in this area is by Pazour & Meller (2014). However, more research is needed, particularly to include more recent automated and robotic systems.

Warehouse Sustainability

Increased social awareness, together with governmental regulations for carbon emissions and waste management, has transformed sustainability from an idealistic idea to an absolute necessity for companies (Chaabane et al., 2011). While increasing attention has focused on supply chain sustainability (e.g., Seuring & Müller (2008); Ballot & Fontane (2010); Barjis et al. (2010); Barber et al. (2012)), the environmental impact of automated warehouses has not received much attention. Colicchia et al. (2011) offer several approaches for more sustainable warehouses, such as using green energy sources, optimizing travel distance and storage assignment policies, and adopting energy-efficient material handling equipment. Tappia et al. (2015) propose a mathematical model to evaluate the energy consumption and environmental impact of AS/R and AVS/R systems. Zaerpour et al. (2015) do a similar analysis for a live-cube storage system. Hahn-Woernle & Günthner (2018) propose a power-load management policy for multi aisle mini-load AS/R systems to reduce energy consumption peaks, thereby lowering the energy cost of the automated warehouse, with only a slight decrease in throughput. However, more studies are needed to incorporate environmental aspects into the decision models revolving around new material handling technologies.

In this dissertation, we touched upon several aspects and challenges regarding the robotized warehouses. However, as mentioned in this chapter, there are still several problems and questions that require attention from researchers. We believe this dissertation may provide a solid foundation to help future academics address these challenges.

Bibliography

ABI Research (2019). Robotics in e-commerce fulfillment. Accessed May 3, 2019 `https://bit.ly/2YwQg3J`.

Agatz, N. A., Fleischmann, M., & Van Nunen, J. A. (2008). E-fulfillment and multi-channel distribution–a review. *European journal of operational research*, 187(2), 339–356.

Akyildiz, I. F. (1988). Mean value analysis for blocking queueing networks. *IEEE Transactions on Software Engineering*, 14(4), 418–428.

Alfieri, A., Cantamessa, M., Monchiero, A., & Montagna, F. (2012). Heuristics for puzzle-based storage systems driven by a limited set of automated guided vehicles. *Journal of Intelligent Manufacturing*, 23(5), 1695–1705.

Automation Parking System (2016). Park, swipe, leave system. Accessed May 30, 2016 `https://bit.ly/3i3HdRR`.

AutoStore (2018). AutoStore: introduction. YouTube. Accessed, September 3, 2018 `https://bit.ly/2YWPJZO`.

Azadeh, K., De Koster, R., & Roy, D. (2019a). Robotized and automated warehouse systems: Review and recent developments. *Transportation Science*, 53(4), 917–945.

Azadeh, K., Roy, D., & De Koster, R. (2019b). Design, modeling, and analysis of vertical robotic storage and retrieval systems. *Transportation Science*, 53(5), 1213–1234.

Ballot, E. & Fontane, F. (2010). Reducing transportation CO2 emissions through pooling of supply networks: Perspectives from a case study in french retail chains. *Production Planning & Control*, 21(6), 640–650.

Banker, S. (2016). *Robots In The Warehouse: It's Not Just Amazon*. Technical report, Forbes.

Banker, S. (2018). Smart mobile robots are everywhere. Forbes. Accessed May 3, 2019 `https://bit.ly/2EflXpR`.

Barber, K. D., Beach, R., & Zolkiewski, J. (2012). Environmental sustainability: a value cycle research agenda. *Production Planning & Control*, 23(2-3), 105–119.

Barjis, J., Verbraeck, A., Tan, K.-S., Daud Ahmed, M., & Sundaram, D. (2010). Sustainable enterprise modelling and simulation in a warehousing context. *Business Process Management Journal*, 16(5), 871–886.

Bartholdi III, J. J. & Platzman, L. K. (1986). Retrieval strategies for a carousel conveyor. *IIE Transactions*, 18(2), 166–173.

Baskett, F., Chandy, K. M., Muntz, R. R., & Palacios, F. G. (1975). Open, closed, and mixed networks of queues with different classes of customers. *Journal of the ACM (JACM)*, 22(2), 248–260.

Bastian Solutions (2016). Personal communication.

Bertsekas, D. (1995). Dynamic programming and optimal control, vol. ii. *Athena Scientific, Bellmont*.

Bolch, G., Greiner, S., de Meer, H., & Trivedi, K. S. (2006). *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. John Wiley & Sons.

Boysen, N., Briskorn, D., & Emde, S. (2017). Parts-to-picker based order processing in a rack-moving mobile robots environment. *European journal of operational research*, 262(2), 550–562.

Boysen, N., De Koster, R., & Füßler, D. (2020). The forgotten sons: Warehousing systems for brick-and-mortar retail chains. *European journal of operational research*. to appear.

Boysen, N., De Koster, R., & Weidinger, F. (2019). Warehousing in the e-commerce era: A survey. *European Journal of Operational Research*, 277(2), 396–411.

Boysen, N. & Stephan, K. (2016). A survey on single crane scheduling in automated storage/retrieval systems. *European Journal of Operational Research*, 254(3), 691–704.

Bozer, Y. A. & White, J. A. (1984). Travel-time models for automated storage/retrieval systems. *IIE Transactions*, 16(4), 329–338.

Brynjolfsson, E., Hu, Y., & Smith, M. D. (2003). Consumer surplus in the digital economy: Estimating the value of increased product variety at online booksellers. *Management Science*, 49(11), 1580–1596.

Buck Consultants International (2017). Robotisering bespaart de komende 15 jaren 35.000 arbeidsplaatsen in nederlandse distributiecentra. Accessed May 30, 2017, `https://bit.ly/3dvSkzp`.

Cai, X., Heragu, S. S., & Liu, Y. (2013). Modeling automated warehouses using semi-open queueing networks. In *Handbook of Stochastic Models and Analysis of Manufacturing System Operations* (pp. 29–71). Springer.

Cai, X., Heragu, S. S., & Liu, Y. (2014). Modeling and evaluating the avs/rs with tier-to-tier vehicles using a semi-open queueing network. *IIE Transactions*, 46(9), 905–927.

Caputo, A. C. & Pelagagge, P. M. (2006). Management criteria of automated order picking systems in high-rotation high-volume distribution centers. *Industrial Management & Data Systems*, 106(9), 1359–1383.

Carlo, H. J. & Vis, I. F. (2012). Sequencing dynamic storage systems with multiple lifts and shuttles. *International Journal of Production Economics*, 140(2), 844–853.

Cavazos-Cadena, R. & Sennott, L. I. (1992). Comparing recent assumptions for the existence of average optimal stationary policies. *Operations research letters*, 11(1), 33–37.

Chaabane, A., Ramudhin, A., & Paquet, M. (2011). Designing supply chains with sustainability considerations. *Production Planning & Control*, 22(8), 727–741.

Chadès, I., Chapron, G., Cros, M.-J., Garcia, F., & Sabbadin, R. (2014). Mdptoolbox: a multi-platform toolbox to solve stochastic dynamic programming problems. *Ecography*, 37(9), 916–920.

Chandy, K. M., Herzog, U., & Woo, L. (1975). Parametric analysis of queuing networks. *IBM Journal of Research and Development*, 19(1), 36–42.

Chen, Y.-L., Tang, K., Shen, R.-J., & Hu, Y.-H. (2005). Market basket analysis in a multiple store environment. *Decision Support Systems*, 40(2), 339 – 354.

Chiou, J.-S., Chou, S.-Y., & Shen, G. C.-C. (2017). Consumer choice of multichannel shopping. *Internet Research*, 77.

Colicchia, C., Melacini, M., & Perotti, S. (2011). Benchmarking supply chain sustainability: insights from a field study. *Benchmarking: An International Journal*, 18(5), 705–732.

D'Antonio, G., De Maddis, M., Bedolla, J. S., Chiabert, P., & Lombardi, F. (2018). Analytical models for the evaluation of deep-lane autonomous vehicle storage and retrieval system performance. *The International Journal of Advanced Manufacturing Technology*, 94(5-8), 1811–1824.

De Koster, R. (1994). Performance approximation of pick-to-belt orderpicking systems. *European Journal of Operational Research*, 72(3), 558–573.

De Koster, R. (2015). *Past and Future. Perspectives on Material Handling.* Rotterdam, Netherlands: Erasmus Research Institute of Management.

De Koster, R., Le-Duc, T., & Roodbergen, K. J. (2007). Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182(2), 481–501.

De Koster, R., Le-Duc, T., & Yugang, Y. (2008). Optimal storage rack design for a 3-dimensional compact AS/RS. *International Journal of Production Research*, 46(6), 1495–1514.

De Koster, R. & Yu, M. (2008). Minimizing makespan and throughput times at Aalsmeer flower auction. *Journal of the Operational Research Society*, 59(9), 1182–1190.

De Koster, R. B., Le-Duc, T., & Zaerpour, N. (2012). Determining the number of zones in a pick-and-sort order picking system. *International Journal of Production Research*, 50(3), 757–771.

De Vries, J., De Koster, R., & Stam, D. (2016). Aligning order picking methods, incentive systems, and regulatory focus to increase performance. *Production and Operations Management*, 25(8), 1363–1376.

De Weerd, P. (2019). Jumbo opent derde e-fulfilment center. Logistiek. Accessed May 18, 2020 `https://bit.ly/3g21eXS`.

Dhingra, V., Kumawat, G. L., Roy, D., & De Koster, R. (2018). Solving semi-open queuing networks with time-varying arrivals: An application in container terminal landside operations. *European Journal of Operational Research*, 267(3), 855–876.

Dijkhuizen, B. (2019a). Dit is het nieuwe omnichannel distributiecentrum van de Bijenkorf. Logistiek. Accessed Dec 12, 2019 `https://bit.ly/2EArNC0`.

Dijkhuizen, B. (2019b). Herinrichting omnichannel dc Blokker in eindfase. Logistiek. Accessed Dec 12, 2019 `https://bit.ly/35GETcI`.

Dijkhuizen, B. (2020). Albert Heijn opent twee nieuwe home shop centers. Logistiek. Accessed May 18, 2020 `https://bit.ly/2Tj1aJp`.

Dijkstra, A. S. & Roodbergen, K. J. (2017). Exact route-length formulas and a storage location assignment heuristic for picker-to-parts warehouses. *Transportation Research Part E: Logistics and Transportation Review*, 102, 38–59.

Dukic, G., Opetuk, T., & Lerher, T. (2015). A throughput model for a dual-tray vertical lift module with a human order-picker. *International Journal of Production Economics*, 170, 874–881.

Ekren, B. Y. (2011). Performance evaluation of avs/rs under various design scenarios: a case study. *The International Journal of Advanced Manufacturing Technology*, 55(9), 1253–1261.

Ekren, B. Y. (2017). Graph-based solution for performance evaluation of shuttle-based storage and retrieval system. *International Journal of Production Research*, 55(21), 6516–6526.

Ekren, B. Y. & Heragu, S. S. (2010a). Approximate analysis of load-dependent generally distributed queuing networks with low service time variability. *European Journal of Operational Research*, 205(2), 381–389.

Ekren, B. Y. & Heragu, S. S. (2010b). Simulation-based regression analysis for the rack configuration of an autonomous vehicle storage and retrieval system. *International Journal of Production Research*, 48(21), 6257–6274.

Ekren, B. Y., Heragu, S. S., Krishnamurthy, A., & Malmborg, C. J. (2010). Simulation based experimental design to identify factors affecting performance of AVS/RS. *Computers & Industrial Engineering*, 58(1), 175–185.

Ekren, B. Y., Heragu, S. S., Krishnamurthy, A., & Malmborg, C. J. (2013). An approximate solution for semi-open queueing network model of an autonomous vehicle storage and retrieval system. *IEEE Transactions on Automation Science and Engineering*, 10(1), 205–215.

Ekren, B. Y., Heragu, S. S., Krishnamurthy, A., & Malmborg, C. J. (2014). Matrix-geometric solution for semi-open queuing network model of autonomous vehicle storage and retrieval system. *Computers & Industrial Engineering*, 68, 78–86.

Ekren, Y. B., Sari, Z., & Rosi, B. (2015). Simulation analysis of shuttle based storage and retrieval systems. *International journal of simulation modelling*, 14(1), 48–59.

Enright, J. J. & Wurman, P. R. (2011). Optimization and coordinated autonomy in mobile fulfillment systems. Automated Action Planning for Autonomous Mobile Robots.

Epp, M., Wiedemann, S., & Furmans, K. (2017). A discrete-time queueing network approach to performance evaluation of autonomous vehicle storage and retrieval systems. *International Journal of Production Research*, 55(4), 960–978.

EWECO (2016). Space parking optimization technology (spot). Accessed May 30, 2016, `https://bit.ly/387NbfX`.

Fukunari, M., Bennett, K. P., & Malmorg, C. J. (2004). Decision-tree learning in dwell point policies in autonomous vehicle storage and retrieval systems (avsrs). In M. Kantardzic, O. Nasraoui, & M. Milanova (Eds.), *Machine Learning and Applications, 2004. Proceedings. 2004 International Conference on* (pp. 81–84). Piscataway, NJ: IEEE.

Fukunari, M. & Malmborg, C. J. (2008). An efficient cycle time model for autonomous vehicle storage and retrieval systems. *International Journal of Production Research*, 46(12), 3167–3184.

Fukunari, M. & Malmborg, C. J. (2009). A network queuing approach for evaluation of performance measures in autonomous vehicle storage and retrieval systems. *European Journal of Operational Research*, 193(1), 152 – 167.

Furmans, K., Nobbe, C., & Schwab, M. (2011). Future of material handling–modular, flexible and efficient. In N. M. Amato, I.-M. Chen, A. De Luca, C. Jenkins, D. Kragic, N. Papanikolopoulos, F. Park, L. Parker, S. Sugano, & F. van der Stappen (Eds.), *IEEE/RSJ International Conference on Intelligent Robots and Systems* San Francisco, USA: IEEE.

Furmans, K., Schönung, F., & Gue, K. R. (2010). Plug and work material handling systems. *Progress in Material Handling Research*, (pp. 132–142).

Füßler, D. & Boysen, N. (2017a). Efficient order processing in an inverse order picking system. *Computers & Operations Research*, 88, 150–160.

Füßler, D. & Boysen, N. (2017b). High-performance order processing in picking workstations. *EURO Journal on Transportation and Logistics*, (pp. 1–26). https://doi.org/10.1007/s13676-017-0113-8.

Füßler, D. & Boysen, N. (2019). High-performance order processing in picking workstations. *EURO Journal on Transportation and Logistics*, 8(1), 65–90.

Gagliardi, J.-P., Renaud, J., & Ruiz, A. (2012). Models for automated storage and retrieval systems: a literature review. *International Journal of Production Research*, 50(24), 7110–7125.

Glover, F. W. & Kochenberger, G. A. (2006). *Handbook of metaheuristics*, volume 57. Springer Science & Business Media.

Goetschalckx, M. & Ratliff, D. H. (1988). Order picking in an aisle. *IIE Transactions*, 20(1), 53–62.

Gue, K. R. (2006). Very high density storage systems. *IIE Transactions*, 38(1), 79–90.

Gue, K. R., Furmans, K., Seibold, Z., & Uludağ, O. (2014). Gridstore: a puzzle-based storage system with decentralized control. *IEEE Transactions on Automation Science and Engineering*, 11(2), 429–438.

Gue, K. R. & Kim, B. S. (2007). Puzzle-based storage systems. *Naval Research Logistics (NRL)*, 54(5), 556–567.

Gue, K. R. & Uludag, O. (2012). A high-density, puzzle-based order picking system. In A. L. Carrano, K. Gue, R. De Koster, M. Ogle, B. Montreuil, & J. Smith (Eds.), *12th International Material Handling Research Colloquium* (pp. 209–220). Charlotte, NC: CICMHE.

Hahn-Woernle, P. & Günthner, W. A. (2018). Power-load management reduces energy-dependent costs of multi-aisle mini-load automated storage and retrieval systems. *International Journal of Production Research*, 56(3), 1269–1285.

Hao, J., Yu, Y., & Zhang, L. L. (2015). Optimal design of a 3d compact storage system with the i/o port at the lower mid-point of the storage rack. *International Journal of Production Research*, 53(17), 5153–5173.

Hausman, W. H., Schwarz, L. B., & Graves, S. C. (1976). Optimal storage assignment in automatic warehousing systems. *Management Science*, 22(6), 629–638.

He, S. & Luo, J. (2009). Deadlock control of autonomous vehicle storage and retrieval systems via coloured timed petri nets and digraph tools. *International Journal of Production Research*, 47(12), 3253–3263.

Heragu, S. S., Cai, X., Krishnamurthy, A., & Malmborg, C. J. (2008). Striving for warehouse excellence. *Industrial Engineer*, 40(12), 43–47.

Heragu, S. S., Cai, X., Krishnamurthy, A., & Malmborg, C. J. (2011). Analytical models for analysis of automated warehouse material handling systems. *International Journal of Production Research*, 49(22), 6833–6861.

Hirnand, R. (2018). Inside hema: A look into the cornerstone of Alibaba's new retail strategy. Medium. Accessed Dec 12, 2019 `https://bit.ly/2rSpXd3`.

Hyundai Elevator Co. LTD. (2016). Hip (hyundai integrated parking system). Accessed May 30, 2016, `https://bit.ly/38OC9sO`.

Imahori, S. & Hase, Y. (2016). Graph-based heuristics for operational planning and scheduling problem in automatic picking system. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, 10(3), JAMDSM0039–JAMDSM0039.

Industrie-forum (2004). Vom gelochten winkelprofil zum vollautomatischen hochregallager. Accessed March 28, 2017, `https://bit.ly/2Vd4Ygl`.

Jia, J. & Heragu, S. S. (2009). Solving semi-open queuing networks. *Operations research*, 57(2), 391–401.

Jünemann, R. (1989). *Materialfluß und Logistik*. Springer Berlin.

Karl, S. (2009). Lecture notes on stochastic modeling I. Colombia University. `https://bit.ly/2xJXt7A`.

Kembro, J. H., Norrman, A., & Eriksson, E. (2018). Adapting warehouse operations and design to omni-channel logistics: A literature review and research agenda. *International Journal of Physical Distribution & Logistics Management*, 48(9), 890–912.

Kim, W. R., Kim, W. R., Kim, N., Kim, N., Chang, Y. S., & Chang, Y. S. (2016). A study on the control of an automatic picking system (aps) in a warehouse. *Assembly Automation*, 36(4), 388–397.

Kota, V. R., Taylor, D., & Gue, K. R. (2015). Retrieval time performance in puzzle-based storage systems. *Journal of Manufacturing Technology Management*, 26(4), 582–602.

Kumar, A., Roy, D., & Tiwari, M. (2014). Optimal partitioning of vertical zones in vehicle-based warehouse systems. *International Journal of Production Research*, 52(5), 1285–1305.

Kumawat, G. L., Roy, D., de Koster, M., & Adan, I. (2018). Stochastic modelling of parallel process flows in intra-logistics systems. Available at SSRN: https://ssrn.com/abstract=3216327 or http://dx.doi.org/10.2139/ssrn.3216327.

Kuo, P.-H., Krishnamurthy, A., & Malmborg, C. J. (2007). Design models for unit load storage and retrieval systems using autonomous vehicle technology and resource conserving storage and dwell point policies. *Applied Mathematical Modelling*, 31(10), 2332–2346.

Kuo, P.-H., Krishnamurthy, A., & Malmborg, C. J. (2008). Performance modelling of autonomous vehicle storage and retrieval systems using class-based storage policies. *International Journal of Computer Applications in Technology*, 31(3-4), 238–248.

Lamballais, T., Merschformann, M., Roy, D., Suhl, L., & De Koster, M. (2017a). *Optimal policies for resource reallocation in a robotic mobile fulfillment system*. Technical report, Working paper.

Lamballais, T., Roy, D., & De Koster, R. (2017b). Estimating performance in a robotic mobile fulfillment system. *European Journal of Operational Research*, 256(3), 976–990.

Lamballais, T., Roy, D., & De Koster, R. (2017c). Inventory allocation in robotic mobile fulfillment systems. *ERIM Report Series*. Available at SSRN: https://ssrn.com/abstract=2900940 or http://dx.doi.org/10.2139/ssrn.2900940.

Larco, J. A., De Koster, R., Roodbergen, K. J., & Dul, J. (2017). Managing warehouse efficiency and worker discomfort through enhanced storage assignment decisions. *International Journal of Production Research*, 55(21), 6407–6422.

Lee, H.-Y. & Murray, C. C. (2019). Robotics in order picking: evaluating warehouse layouts for pick, place, and transport vehicle routing systems. *International Journal of Production Research*, 57(18), 5821–5841.

Lerher, T. (2016). Travel time model for double-deep shuttle-based storage and retrieval systems. *International Journal of Production Research*, 54(9), 2519–2540.

Lerher, T., Borovinsek, M., Ficko, M., & Palcic, I. (2017). Parametric study of throughput performance in sbs/rs based on simulation. *International Journal of Simulation Modelling*, 16(1), 96–107.

Lerher, T., Ekren, B. Y., Dukic, G., & Rosi, B. (2015). Travel time model for shuttle-based storage and retrieval systems. *The International Journal of Advanced Manufacturing Technology*, 78(9-12), 1705–1725.

Lerher, T., Ekren, B. Y., Sari, Z., & Rosi, B. (2016). Method for evaluating the throughput performance of shuttle based storage and retrieval systems. *Tehnicki vjesnik/Technical Gazette*, 23(3).

Lerher, T., Potrč, I., Šraml, M., & Tollazzi, T. (2010). Travel time models for automated warehouses with aisle transferring storage and retrieval machine. *European Journal of Operational Research*, 205(3), 571–583.

Lippman, S. A. (1975). Applying a new device in the optimization of exponential queuing systems. *Operations Research*, 23(4), 687–710.

Litvak, N. & Vlasiou, M. (2010). A survey on performance analysis of warehouse carousel systems. *Statistica Neerlandica*, 64(4), 401–447.

LocusBots (2018). Simply pick faster - Locus Robotics overview. YouTube. Accessed September 3, 2018 `https://bit.ly/2No6Kqq`.

Löffler, M., Boysen, N., & Schneider, M. (2018). *Picker routing in AGV-assisted order picking systems*. Technical report, Working Paper, DPO-01/2018, Deutsche Post Chair-Optimization of Distribution.

Loyd, S. & Gardner, M. (1959). *Mathematical puzzles*, volume 1. Courier Corporation.

Malmborg, C. J. (2002). Conceptualizing tools for autonomous vehicle storage and retrieval systems. *International Journal of Production Research*, 40(8), 1807–1822.

Malmborg, C. J. (2003a). Design optimization models for storage and retrieval systems using rail guided vehicles. *Applied Mathematical Modelling*, 27(12), 929–941.

Malmborg, C. J. (2003b). Interleaving dynamics in autonomous vehicle storage and retrieval systems. *International Journal of Production Research*, 41(5), 1057–1069.

Manzini, R., Accorsi, R., Baruffaldi, G., Cennerazzo, T., & Gamberi, M. (2016). Travel time models for deep-lane unit-load autonomous vehicle storage and retrieval system (avs/rs). *International Journal of Production Research*, 54(14), 4286–4304.

Marchet, G., Melacini, M., Perotti, S., & Tappia, E. (2012). Analytical model to estimate performances of autonomous vehicle storage and retrieval systems for product totes. *International Journal of Production Research*, 50(24), 7134–7148.

Marchet, G., Melacini, M., Perotti, S., & Tappia, E. (2013). Development of a framework for the design of autonomous vehicle storage and retrieval systems. *International Journal of Production Research*, 51(14), 4365–4387.

Meisel, F. & Bierwirth, C. (2013). A framework for integrated berth allocation and crane operations planning in seaport container terminals. *Transportation Science*, 47(2), 131–147.

Melacini, M., Perotti, S., & Tumino, A. (2011). Development of a framework for pick-and-pass order picking system design. *The International Journal of Advanced Manufacturing Technology*, 53(9-12), 841–854.

Meller, R. D. & Klote, J. F. (2004). A throughput model for carousel/VLM pods. *IIE Transactions*, 36(8), 725–741.

Meller, R. D., Nazzal, D., & Thomas, L. M. (2018). Collaborative bots in distribution centers. In E. Ballot, A. L. Carrano, K. Ellis, W. Farrell, J. Porter, & D. Reed (Eds.), *15th International Material Handling Research Colloquium* Charlotte, NC: CICMHE.

Meller, R. D. & Pazour, J. A. (2008). A heuristic for sku assignment and allocation in an a-frame system. In J. Fowler & S. Mason (Eds.), *IIE Annual Conference. Proceedings* (pp. 770). Peachtree Corners, GA: Institute of Industrial Engineers-Publisher.

MHI (2015). A-frame. Accessed December 06 , 2017 `http://www.mhi.org/solutions-community/solutions-guide/aframe`.

MHI (2015). Vertical lift modules. Accessed January 20 , 2017 `https://bit.ly/2A49jew`.

MHI (2017). Storage equipment. Accessed on January 25, 2018 `https://bit.ly/2Z5qbdi`.

Mirzaei, M., De Koster, R. B., & Zaerpour, N. (2017). Modelling load retrievals in puzzle-based storage systems. *International Journal of Production Research*, 55(21), 6423–6435.

Mirzaei, M., Zaerpour, N., & De Koster, R. (2018). An integrated custer-based storage assignment policy. Working Paper.

Mountz, M. C., D'andrea, R., LaPlante, J. A., David, P. L. I., Mansfield, P. K., & Amsbury, B. W. (2008). Inventory system with mobile drive unit and inventory holder. US Patent 7,402,018.

MWPVL International (2012). Is Kiva Systems a good fit for your distribution center? An unbiased distribution consultant evaluation. Accessed May 30, 2016, `http://www.mwpvl.com/html/kiva_systems.html`.

MWPVL International (2013). *Dematic Multishuttle 2, An In-depth and Independent Review from a Supply Chain Consultant*. Technical report.

Nigam, S., Roy, D., De Koster, R., & Adan, I. (2014). Analysis of class-based storage strategies for the mobile shelf-based order pick system. In J. Smith, K. Ellis, R. De Koster, S. Lavender, B. Montreuil, & M. Ogle (Eds.), *Progress in Material Handling Research* (pp. 1–9). Charlotte, NC: CICMHE.

Ocado (2017). Ocado technology. Accessed December 19 , 2017 `https://bit.ly/2NqT1ze`.

Open Access Government (2019). High street crisis - is it possible to avoid the demise of retailers? Open Access Government. Accessed Dec 12, 2019 `https://bit.ly/2sJLvs4`.

OPEX (2016). Perfect pick is streamlined ecommerce order fulfillment. Accessed May 30, 2016 `https://bit.ly/2YtufF2`.

Pazour, J. A. & Meller, R. D. (2011). An analytical model for a-frame system design. *IIE Transactions*, 43(10), 739–752.

Pazour, J. A. & Meller, R. D. (2013). The impact of batch retrievals on throughput performance of a carousel system serviced by a storage and retrieval machine. *International Journal of Production Economics*, 142(2), 332–342.

Pazour, J. A. & Meller, R. D. (2014). A framework and analysis to inform the selection of piece-level order-fulfillment technologies. In J. Smith, K. Ellis, R. De Koster, S. Lavender, B. Montreuil, & M. Ogle (Eds.), *Progress in Material Handling Research* (pp. 1–25). Charlotte, NC: CICMHE.

Perros, H. G. (1994). *Queueing networks with blocking*. Oxford University Press, Inc.

Pick-n-Go (2010). Coop Pick-n-Go. YouTube. Accessed September 3, 2018 `https://bit.ly/31985K0`.

Puchinger, J. & Raidl, G. R. (2005). Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In M. J. Álvarez JR (Ed.), *International Work-Conference on the Interplay Between Natural and Artificial Computation* (pp. 41–53). Berlin: Springer.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Qimarox BV (2013). Prorunner mk5 vertical conveyor. `https://www.youtube.com/watch?v=e1uvFjKUZz0`.

Ramtin, F. & Pazour, J. A. (2014). Analytical models for an automated storage and retrieval system with multiple in-the-aisle pick positions. *IIE Transactions*, 46(9), 968–986.

Ramtin, F. & Pazour, J. A. (2015). Product allocation problem for an AS/RS with multiple in-the-aisle pick positions. *IIE Transactions*, 47(12), 1379–1396.

Reiser, M. & Lavenberg, S. S. (1980). Mean-value analysis of closed multichain queuing networks. *Journal of the ACM (JACM)*, 27(2), 313–322.

Roodbergen, K. J. & De Koster, R. (2001). Routing methods for warehouses with multiple cross aisles. *International Journal of Production Research*, 39(9), 1865–1883.

Roodbergen, K. J. & Vis, I. F. (2009). A survey of literature on automated storage and retrieval systems. *European Journal of Operational Research*, 194(2), 343–362.

Rosenblatt, M. J. & Eynan, A. (1989). Deriving the optimal boundaries for class-based automatic storage/retrieval systems. *Management Science*, 35(12), 1519–1524.

Rosi, B., Grasic, L., Dukic, G., Opetuk, T., & Lerher, T. (2016). Simulation-based performance analysis of automated single-tray vertical lift module. *International Journal of Simulation Modelling*, 15(1), 97–108.

Roy, D. (2011). *Design and Analysis of Unit-Load Warehouse Operations using Autonomous Vehicles*. PhD thesis, University of Wisconsin - Madison.

Roy, D. (2016). Semi-open queuing networks: a review of stochastic models, solution methods and new research areas. *International Journal of Production Research*, 54(6), 1735–1752.

Roy, D. (2020). Warehousing protocols must change with COVID-19; vital suggestions. International Business Times. Accessed June 2, 2020 `https://bit.ly/2U1S9ER`.

Roy, D., Krishnamurthy, A., Heragu, S. S., & Malmborg, C. (2017). A multi-tier linking approach to analyze performance of autonomous vehicle-based storage and retrieval systems. *Computers & Operations Research*, 83, 173–188.

Roy, D., Krishnamurthy, A., Heragu, S. S., & Malmborg, C. J. (2012). Performance analysis and design trade-offs in warehouses with autonomous vehicle technology. *IIE Transactions*, 44(12), 1045–1060.

Roy, D., Krishnamurthy, A., Heragu, S. S., & Malmborg, C. J. (2014). Blocking effects in warehouse systems with autonomous vehicles. *IEEE Transactions on Automation Science and Engineering*, 11(2), 439–451.

Roy, D., Krishnamurthy, A., Heragu, S. S., & Malmborg, C. J. (2015a). Queuing models to analyze dwell-point and cross-aisle location in autonomous vehicle-based warehouse systems. *European Journal of Operational Research*, 242(1), 72–87.

Roy, D., Krishnamurthy, A., Heragu, S. S., & Malmborg, C. J. (2015b). Stochastic models for unit-load operations in warehouse systems with autonomous vehicles. *Annals of Operations Research*, 231(1), 129–155.

Roy, D., Krishnamurthy, A., Heragu, S. S., & Malmborg, C. J. (2016). A simulation framework for studying blocking effects in warehouse systems with autonomous vehicles. *European Journal of Industrial Engineering*, 10(1), 51–80.

Roy, D., Nigam, S., de Koster, R., Adan, I., & Resing, J. (2019a). Robot-storage zone assignment strategies in mobile fulfillment systems. *Transportation Research Part E: Logistics and Transportation Review*, 122, 119 – 142.

Roy, D., Nigam, S., de Koster, R., Adan, I., & Resing, J. (2019b). Robot-storage zone assignment strategies in mobile fulfillment systems. *Transportation Research Part E: Logistics and Transportation Review*, 122, 119–142.

Sadowsky, V. & Ten Hompel, M. (2011). Calculation of the average travel distance in a low-level picker-to-part system considering any distribution function within the aisles. *Logistics Journal*, 2011(01).

Saghiri, S., Wilding, R., Mena, C., & Bourlakis, M. (2017). Toward a three-dimensional framework for omni-channel. *Journal of Business Research*, 77, 53–67.

Sari, Z., Saygin, C., & Ghouali, N. (2005). Travel-time models for flow-rack automated storage and retrieval systems. *The International Journal of Advanced Manufacturing Technology*, 25(9-10), 979–987.

Schaverien, A. (2018). Five reasons why Amazon is moving into Bricks-And-Mortar retail. Forbes. Accessed Dec 12, 2019 `https://bit.ly/2Z2sbSY`.

Schwerdfeger, S. & Boysen, N. (2017). Order picking along a crane-supplied pick face: The sku switching problem. *European Journal of Operational Research*, 260(2), 534–545.

Seuring, S. & Müller, M. (2008). From a literature review to a conceptual framework for sustainable supply chain management. *Journal of Cleaner Production*, 16(15), 1699–1710.

Skrovan, S. (2017). Why researching online, shopping offline is the new norm. Retail Dive. Accessed May 1, 2020 `https://bit.ly/2yZoy7C`.

Stadtler, H. (1996). An operational planning concept for deep lane storage systems. *Production and Operations Management*, 5(3), 266–282.

Tappia, E., Marchet, G., Melacini, M., & Perotti, S. (2015). Incorporating the environmental dimension in the assessment of automated warehouses. *Production Planning & Control*, 26(10), 824–838.

Tappia, E., Roy, D., De Koster, R., & Melacini, M. (2016). Modeling, analysis, and design insights for shuttle-based compact storage systems. *Transportation Science*, 51(1), 269–295.

Taylor, G. D. & Gue, K. R. (2008). The effects of empty storage locations in puzzle-based storage systems. In *IIE Annual Conference. Proceedings* (pp. 519). Peachtree Corners, GA: Institute of Industrial Engineers-Publisher.

TORU (2017). Pick-by-Robot: warehouse automation with Magazino at FIEGE Logistik. YouTube. Accessed September 3, 2018 `https://bit.ly/3exWkki`.

Trebilcock, B. (2018). Nextgen supply chain at dhl. Accessed November 20, 2019, `https://bit.ly/31eFF18`.

Uichanco, J., Jasin, S., & Roy, D. (2019). Chasing the e-commerce market: The omni-channel revolution. Open Access Government. Accessed December 10, 2019 `https://bit.ly/2YxHnJb`.

Uludag, O. (2014). *GridPick: A High Density Puzzle Based Order Picking System with Decentralized Control*. PhD thesis, Auburn University.

Van der Gaast, J. P., De Koster, R. B., Adan, I. J., & Resing, J. A. (2020). Capacity analysis of sequential zone picking systems. *Operations Research*, 68(1), 161–179.

Van Dijk, N. M. (1988). On jackson's product form with jump-over blocking. *Operations research letters*, 7(5), 233–235.

Van Doremalen, J. (1986). *Approximate analysis of queueing network models.* PhD thesis, Department of Mathematics and Computer Science.

Van Doremalen, J. & Wessels, J. (1986). Iterative approximations for networks of queues. In *Stochastic programming* (pp. 95–106). Springer.

Van Gils, T., Ramaekers, K., Braekers, K., Depaire, B., & Caris, A. (2018). Increasing order picking efficiency by integrating storage, batching, zone picking, and routing policy decisions. *International Journal of Production Economics*, 197, 243–261.

Walrand, J. (1983). A note on Norton's theorem for queuing networks. *Journal of Applied Probability*, (pp. 442–444).

Whitt, W. (1983). The queueing network analyzer. *Bell Labs Technical Journal*, 62(9), 2779–2815.

Wöhr (2016). Wöhr parksafe. Accessed May 30, 2016, `https://bit.ly/38OMViv`.

Wurman, P. R., D'Andrea, R., & Mountz, M. (2008). Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI Magazine*, 29(1), 9.

Yang, P., Miao, L., Xue, Z., & Qin, L. (2015). Optimal storage rack design for a multi-deep compact as/rs considering the acceleration/deceleration of the storage and retrieval machine. *International Journal of Production Research*, 53(3), 929–943.

Yu, H., Yu, Y., De Koster, R., & Zaerpour, N. (2017). Optimal algorithm for minimizing retrieval time in puzzle-based storage system with multiple simultaneously movable empty cells. *Working Paper*.

Yu, M. & De Koster, R. (2008). Performance approximation and design of pick-and-pass order picking systems. *IIE Transactions*, 40(11), 1054–1069.

Yu, M. & De Koster, R. (2010). Enhancing performance in order picking processes by dynamic storage systems. *International Journal of Production Research*, 48(16), 4785–4806.

Yu, M. & De Koster, R. B. (2009a). The impact of order batching and picking area zoning on order picking system performance. *European Journal of Operational Research*, 198(2), 480–490.

Yu, Y. & De Koster, R. (2009b). Designing an optimal turnover-based storage rack for a 3D compact automated storage and retrieval system. *International Journal of Production Research*, 47(6), 1551–1571.

Yu, Y. & De Koster, R. (2009c). Optimal zone boundaries for two-class-based compact three-dimensional automated storage and retrieval systems. *IIE Transactions*, 41(3), 194–208.

Yu, Y. & De Koster, R. (2012). Sequencing heuristics for storing and retrieving unit loads in 3d compact automated warehousing systems. *IIE Transactions*, 44(2), 69–87.

Yu, Y., De Koster, R., & Guo, X. (2015). Class-based storage with a finite number of items: Using more classes is not always better. *Production and Operations Management*, 24(8), 1235–1247.

Yuan, R., Graves, S. C., & Cezik, T. (2019). Velocity-based storage assignment in semi-automated storage systems. *Production and Operations Management*, 28(2), 354–373.

Yuan, Z. & Gong, Y. Y. (2017). Bot-in-time delivery for robotic mobile fulfillment systems. *IEEE Transactions on Engineering Management*, 64(1), 83–93.

Zaerpour, N., De Koster, R., & Yu, Y. (2013). Storage policies and optimal shape of a storage system. *International Journal of Production Research*, 51(23-24), 6891–6899.

Zaerpour, N., Yu, Y., & De Koster, R. (2017a). Optimal two-class-based storage in a live-cube compact storage system. *IISE Transactions*, 49(7), 653–668.

Zaerpour, N., Yu, Y., & De Koster, R. (2017b). Response time analysis of a live-cube compact storage system with two storage classes. *IISE Transactions*, 49(5), 461–480.

Zaerpour, N., Yu, Y., & De Koster, R. B. (2015). Small is beautiful: A framework for evaluating and optimizing live-cube compact storage systems. *Transportation Science*.

Zhang, L., Krishnamurthy, A., Malmborg, C. J., & Heragu, S. S. (2009). Variance-based approximations of transaction waiting times in autonomous vehicle storage and retrieval systems. *European Journal of Industrial Engineering*, 3(2), 146–169.

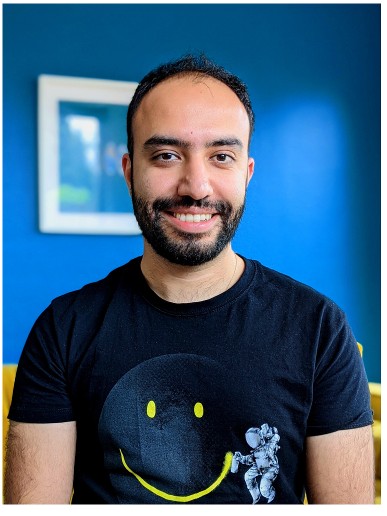
Zijm, W. (2002). Manufacturing and logistic systems analysis, planning and control. *Universiteit Twente*.

Zou, B., De Koster, R., & Xu, X. (2018a). Operating policies in robotic compact storage and retrieval systems. *Transportation Science*, 52(4), 788–811.

Zou, B., Xu, X., Gong, Y., & De Koster, R. (2016). Modeling parallel movement of lifts and vehicles in tier-captive vehicle-based warehousing systems. *European Journal of Operational Research*, 254(1), 51 – 67.

Zou, B., Xu, X., Gong, Y. Y., & Koster, R. D. (2018b). Evaluating battery charging and swapping strategies in a robotic mobile fulfillment system. *European Journal of Operational Research*, 267(2), 733 – 753.

About the author



Kaveh Azadeh was born in Tehran (Iran) on March 21, 1988. He studied Electrical Engineering at the University of Tehran in Iran. In May 2013, he received his M.Sc. degree in Industrial Engineering from the University of Central Florida (UCF) in the USA. He then spent a year as a lead researcher in the Logistics Delivery System Design Laboratory at UCF. In 2014, Kaveh joined the department of Technology and Operations Management at Rotterdam School of Management, Erasmus University, under the supervision of Professor René De Koster and Dr. Debjit Roy.

Kaveh's research interests include the design, analysis, and optimization of intra-logistic systems focusing on the stochastic modeling and the performance evaluation of various robotized order picking systems. His works have been published in Transportation Science and have been presented in several international conferences, including INFORMS Annual Meeting, European Conference on Operational Research, POMS Annual Meeting, IFORS Conference, International Conference on Computational Logistics, and International Conference on Logistics and Maritime Systems. He has also served as a reviewer of various journals, including Transportation Science and OR Spectrum.

Portfolio

Publications

Publications in Journals:

Bodnar, P., de Koster, R., and Azadeh, K. (2017). Scheduling trucks in a cross-dock with mixed service mode dock doors. *Transportation Science*, 51(1), 112-131.

Azadeh, K., Roy, D. and De Koster, R., (2019). Design, modeling, and analysis of vertical robotic storage and retrieval systems. *Transportation Science*, 53(5):1213-1234.

Azadeh, K., De Koster, R., and Roy, D. (2019). Robotized and automated warehouse systems: Review and recent developments. *Transportation Science*, 53(4):917-945.

Conference Papers:

Azadeh, K., Roy, D. and De Koster, R., (2019). Dynamic cobot order picking strategies for a pick-support AGV system. In H.E. Romeijn, A. Schaefer, R. Thomas (Eds.), *IISE Annual Conference. Proceedings* (pp. 1183). Peachtree Corners, GA: Institute of Industrial Engineers-Publisher.

Working Papers:

Azadeh, K., Roy, D., and De Koster, R. (2020). Dynamic human-robot collaborative picking strategies. *Available at SSRN 3585396.*

PhD Courses

Strategic Behavior in Service Systems	Applied Econometrics
Algorithms and Complexity	Scientific Integrity
Integer Programming Methods	Publishing Strategy
Noncooperative Games	Transport Logistics Modelling
Markov Decision Processes	Public Transport
Facility Logistics Management	Programming
Introduction to Data Visualization, Web Scraping, and Text Analysis in R	Stated Preference Data collection & Discrete Choice Modelling

Teaching

Lecturer:

Research Training and Bachelor Thesis 2015 2019-2020
Mathematics 2019

Tutorial Lecturer:

Operations Management 2016-2019

Guest Lecturer:

Facility Logistics Management 2016-2020

Conferences Attendance and Invited Sessions

ICCL 2015, Delft, The Netherlands
INFORMS 2015, Philadelphia, USA
TSL Workshop 2016, Atlanta, USA
ELA Workshop 2017, Wroclaw, Poland
IFORS 2017, Québec City, Canada
INFORMS 2017, Huoston, USA
CEMS Research Seminar 2018, Riezlern, Austria
EURO 2018, Valencia, Spain
OML National Conference 2018, Soesterberg, The Netherlands
CEMS Research Seminar 2019, Riezlern, Austria
Logistics of Autonomous Vessels Workshop 2019, Bergen, Norway
LOGMS 2019, Singapoure

Summary

Different customers may prefer different sales channels and product delivery options. Some customers prefer the online channel (e-commerce) since it provides them with flexibility in the time of ordering. Other might prefer offline channels (physical stores) since they can get instant satisfaction from immediate possession of the purchased products. Some consumers combine the two shopping experiences by browsing for the items online while making the actual purchase at a physical store, or the other way around. Therefore, to stay competitive, most retailers now offer both online and offline channels to their customers.

Different types of warehouses have emerged as a result of these multi-channel retail operations, each with different requirements and objectives. In an offline channel, warehouses act as a distribution center for store replenishment. These warehouses fulfill orders with many lines with large volume per line, from a medium to large assortment of products under moderate time pressure. On the other hand, in an online channel, orders are shipped directly from a warehouse to the customers. A typical e-commerce warehouse fulfills small-sized orders, from a large assortment of products, under significant time pressure, and needs to be flexible enough to adjust to unpredictable demand fluctuations. Duo to the inherent difference between the two types of warehouses, particularly with respect to the order portfolio, different considerations should be taken into account when improving the efficiency of the fulfillment operation within the warehouse.

Among the warehouse activities, order picking is the most laborious and expensive process. It includes collecting the right amount of the right products for a given set of customer orders. Order picking tasks are often repetitive and suffer from poor ergonomics. Furthermore, finding high-quality labor has become difficult. Therefore, order picking has become the primary candidate for automation to improve efficiency in the fulfillment process. There is no one-size-fits-all solution for warehouse automation, and depending on the warehouse type, different automated systems should be considered.

However, warehouse automation requires considerable scale and a long-term vision, as the investments can be earned back only in the medium and longer-term. Therefore, it is crucial to develop tools to help decision-makers find the correct automated solutions for their warehouses. In this dissertation, we provide useful academic and practical insights by modeling and optimizing the performance of different automated and robotic picking systems for different warehouse types.

Chapter 2 presents an overview of the recent trends in automated warehousing, particularly the use of robotic technologies for order fulfillment. Academic studies have developed several models to optimize the performance of the various automated systems, focusing on short- and long-term decisions. We present these different modeling techniques and their corresponding solution approaches to evaluate the performance of the automated systems. We then provide a decision-making framework that includes different objectives, the corresponding decision variables, and the suitable modeling approaches to address them. Next, we describe well-established automation technologies, including automated storage and retrieval systems with cranes or automated forklifts, carousels, vertical lift modules, automated dispensing systems, aisle-based shuttle systems, grid-based shuttle systems, and robotic mobile fulfillment systems. We review the literature related to the various design and control problems in these systems. Next, we identify promising new emerging technologies that, to date, have hardly received any research attention, including vertical and diagonal robotic storage and retrieval systems, robot-based compact storage and retrieval systems, GridSort, and Pick-Support AMRs. We conclude this chapter by summarizing the unaddressed research questions in established systems and pose research questions for emerging technologies.

Chapter 3 investigates the vertical storage and retrieval system, an emerging robotic technology for e-commerce warehouses. We build closed queuing network models that, in turn, are used to optimize the design of the system. The results show that the optimal height-to-width ratio of a vertical system is around one. Because a large number of system robots may lead to blocking and delays, we compare the effects of different robot blocking protocols on the system throughput: robot Recirculation and Wait-on-Spot. The Wait-on-Spot policy produces a higher system throughput when the number of robots in the system is small. However, for a large number of robots in the system, the Recirculation policy dominates the Wait-on-Spot policy. Finally, we compare the operational costs of the vertical and horizontal systems. The results show that in almost all scenarios, the vertical system produces a similar or higher system throughput with a lower operating cost compared with the horizontal system.

Chapter 4 studies systems in which AMRs collaborates with a human picker to efficiently pick the orders by reducing the pickers' unproductive walking time. Picker travel time can be reduced even more by zoning the storage system, where robots take care of the travel between these zones. However, in an omni-channel warehouse, the optimal zoning strategy for these robotic systems is not clear: few zones are particularly suitable for the large store orders, while many zones are particularly suitable for the small online orders. Therefore, we study the effect of dynamic zoning strategies, i.e., dynamic switching between a No Zoning strategy and a Progressive Zoning strategy. We solve the problem in two stages. First, we develop queuing network models to obtain load-dependent pick throughput rates corresponding to a given number of AMRs and a picking strategy with a fixed number of zones. Then, we develop a Markov-decision model to investigate how higher pick performance can be achieved by dynamically switching between these pick strategies. Using data from an omni-channel warehouse that processes various order sizes, we show that a Dynamic Switching policy can lower operational costs by up to 7 percent. However, these cost savings decrease as the number of robots per picker increases.

In the concluding chapter, we summarize the key results and insights from our research and discuss several directions for future studies. In particular, modular system analysis, human-machine interaction, and warehouse sustainability are areas that require more attention from researchers.

Samenvatting (Summary in Dutch)

Klanten hebben verschillende voorkeuren hoe ze producten willen bestellen en geleverd willen krijgen. Sommige klanten geven de voorkeur aan online kanalen (e-commerce) omdat ze op die manier op een zelfgekozen tijd kunnen bestellen. Anderen geven de voorkeur aan offline kanalen (fysieke winkels) omdat ze het product dan onmiddellijk in bezit kunnen krijgen. Sommige consumenten combineren deze twee winkelervaringen door de producten eerst online te bekijken, waarna ze de daadwerkelijke aankoop in een fysieke winkel doen, of net andersom. Om concurrerend te blijven, bieden veel detailhandelaren daarom nu zowel online als offline (omnichannel) verkoopkanalen aan hun klanten aan.

Verschillende verkoopkanalen leiden ook tot verschillende fysieke distributiekanalen, die verschillende soorten magazijnen vereisen met eigen systemen en doelstellingen. Bij een offline kanaal fungeren magazijnen als een distributiecentrum voor het aanvullen van de winkelvoorraad. Deze magazijnen verwerken planmatig winkelorders met elk veel orderlijnen en een groot volume per lijn, en hebben veelal een middelgroot tot groot productassortiment. Aan de andere kant worden de orders in een online kanaal rechtstreeks vanuit een magazijn naar de klanten verzonden. Een doorsnee e-commerce-magazijn levert kleine orders uit een groot productassortiment onder aanzienlijke tijdsdruk en moet flexibel genoeg zijn om zich aan te passen aan onvoorspelbare schommelingen in de vraag. De fundamentele verschillen tussen deze twee soorten magazijnen, met name in de aard van de orders, hebben geleid tot verschillende systeemkeuzes. Daarom moet in elk magazijn rekening worden gehouden met zulke kenmerken bij het verbeteren van de efficiëntie van het orderverzamel- en uitleverproces.

Van alle activiteiten in het magazijn is orderpicken het meest arbeidsintensieve en kostbare proces. Dit omvat het verzamelen van de juiste hoeveelheid van de juiste producten voor een bepaald aantal klantorders. Orderpicken is vaak repetitief en niet erg ergonomisch. Bovendien is het moeilijk geworden om voldoende goede arbeidskrachten te vinden. Daarom is het orderpickproces de belangrijkste kandidaat voor automatisering om de efficiëntie van het magazijn te vergroten. Er is echter niet een eenduidige automatiseringsoplossing voor magazijnen. Afhankelijk van het soort magazijn moeten verschillende geautomatiseerde systemen in overweging worden genomen.

Magazijnautomatisering vereist een aanzienlijke schaalgrootte en een langetermijnvisie, omdat de investeringen alleen op de middellange en langere termijn kunnen worden terugverdiend. Daarom is het van essentieel belang om hulpmiddelen te ontwikkelen waarmee besluitvormers de juiste geautomatiseerde oplossingen voor hun magazijnen kunnen vinden. Door de prestaties van verschillende geautomatiseerde en gerobotiseerde orderpicksystemen voor verschillende soorten magazijnen te modelleren, te optimaliseren en te vergelijken, bieden we in dit proefschrift waardevolle academische en praktische inzichten.

Hoofdstuk 2 geeft een overzicht van de recente trends op het gebied van magazijnautomatisering, met name het gebruik van robottechnologie voor de verwerking van orders. In wetenschappelijk onderzoek zijn verschillende modellen ontwikkeld voor het optimaliseren van de prestaties van verschillende geautomatiseerde systemen, waarbij de nadruk ligt op korte- en langetermijnbeslissingen. We bespreken deze verschillende modelleertechnieken en de bijbehorende oplossingsbenaderingen om de prestaties van de geautomatiseerde systemen te evalueren. Vervolgens presenteren we een raamwerk voor het nemen van beslissingen over het modelleren van automatisering. Het raamwerk bevat de doelstellingskeuze, de bijbehorende beslissingsvariabelen en de geschikte modelleermethode. Vervolgens beschrijven we reeds gevestigde automatiseringstechnologieën, zoals geautomatiseerde opslag- en ophaalsystemen met kranen of geautomatiseerde magazijntrucks, carrousels, verticale liftmodules, dispensingsystemen, shuttlesystemen met shuttles per gangpad of werkend in een raster en zogenaamde robotic mobile fulfilment systemen. We bespreken de literatuur over de verschillende ontwerp- en besturingsproblemen die zich bij deze systemen voordoen. Vervolgens brengen we veelbelovende nieuwe opkomende technologieën in kaart waar tot nu toe weinig onderzoek naar is gedaan, zoals verticale en diagonale gerobotiseerde opslag- en ophaalsystemen, compacte robotgebaseerde opslag- en uitslagsystemen, GridSort en AMRs (Autonomous Mobile Robots) voor de ondersteuning van het orderpickproces. We sluiten dit hoofdstuk af met een samenvatting van nog onbeantwoorde onderzoeksvragen met betrekking tot bestaande systemen en het formuleren van nieuwe onderzoeksvragen over opkomende technologieën.

Hoofdstuk 3 behandelt zogenaamde verticale gerobotiseerde opslag- en uitslagsystemen, een opkomende robottechnologie voor e-commerce-magazijnen. We modelleren het systeem met behulp van gesloten wachtrijnetwerken. Deze kunnen geanalyseerd worden en gebruikt om het ontwerp van het systeem te optimaliseren. De resultaten laten zien dat de optimale hoogte-breedteverhouding van een verticaal systeem ongeveer één is. Omdat een groot aantal systeemrobots kan leiden tot blokkering en vertragingen, vergelijken we de effecten van verschillende robotblokkeringsprotocollen op de systeemdoorzet: Recirculation en Wait-on-Spot. Bij een klein aantal robots in het systeem zorgt de Wait-on-Spot policy voor een grotere systeemdoorzet. Wanneer het aantal robots in het systeem groot is, presteert de Recirculation policy echter beter dan Wait-on-Spot. Tot slot vergelijken we de operationele kosten van de verticale en traditionele horizontale shuttlesystemen. De resultaten laten zien dat het verticale systeem in bijna alle scenarios een vergelijkbare of grotere systeemdoorzet oplevert met lagere operationele kosten in vergelijking met het horizontale systeem.

In hoofdstuk 4 worden systemen onderzocht waarin AMRs samenwerken met een menselijke orderpicker voor een efficiënte verzameling van orders door de niet-productieve looptijd van de orderpicker te verkorten. De looptijd van orderpickers kan zelfs nog verder worden verminderd door het opslagsysteem in zones onder te verdelen, waarbij robots de verplaatsing tussen deze zones voor hun rekening nemen. In een omni-channel-magazijn is echter niet duidelijk welke zone-strategie optimaal is voor deze robotsystemen: een klein aantal zones is vooral geschikt voor grote winkelorders, terwijl een groot aantal zones vooral geschikt is voor kleine online orders. Om deze reden onderzoeken we het effect van dynamische zone-strategieën, d.w.z. dynamisch schakelen tussen een No Zoning-strategie en een Progressive Zoning-strategie. We lossen het probleem op in twee stadia. Eerst ontwikkelen we modellen voor wachtrijnetwerken om de werklastafhankelijke doorzet van het orderpickproces te berekenen voor een gegeven aantal AMRs en voor een vastgelegd aantal zones. Vervolgens ontwikkelen we een Markov-beslismodel om te onderzoeken op welke manier betere orderpickprestaties kunnen worden behaald door dynamisch te schakelen tussen orderpick-strategieën met verschillend aantal zones. Aan de hand van gegevens van een omni-channel-magazijn dat orders van verschillende grootte verwerkt, laten we zien dat een beleid van dynamisch schakelen (Dynamic Switching) de operationele kosten tot 7 procent kan verlagen. Deze kostenbesparingen nemen echter af naarmate het aantal robots per orderpicker toeneemt.

In het afsluitende hoofdstuk vatten we de belangrijkste resultaten en inzichten uit ons onderzoek samen en behandelen we verschillende richtingen voor toekomstig onderzoek. Onderzoekers zouden vooral meer aandacht moeten besteden aan modulaire systeemanalyse, de interactie tussen mens en machine en de duurzaamheid van magazijnen.

ERIM Ph.D. Series Research in Management

The ERIM PhD Series contains PhD dissertations in the field of Research in Management defended at Erasmus University Rotterdam and supervised by senior researchers affiliated to the Erasmus Research Institute of Management (ERIM). All dissertations in the ERIM PhD Series are available in full text through the ERIM Electronic Series Portal: `http://repub.eur.nl/pub`. ERIM is the joint research institute of the Rotterdam School of Management (RSM) and the Erasmus School of Economics (ESE) at the Erasmus University Rotterdam (EUR).

Dissertations in the last four years

Ahmadi, S., *A Motivational Perspective to Decision-Making and Behavior in Organizations*, Promotors: Prof. J.J.P. Jansen & Dr T.J.M. Mom, EPS-2019-477-S&E, `https://repub.eur.nl/pub/116727`

Akemu, O., *Corporate Responses to Social Issues: Essays in Social Entrepreneurship and Corporate Social Responsibility*, Promotors: Prof. G.M. Whiteman & Dr S.P. Kennedy, EPS-2017-392-ORG, `https://repub.eur.nl/pub/95768`

Albuquerque de Sousa, J.A., *International Stock Markets: Essays on the Determinants and Consequences of Financial Market Development*, Promotors: Prof. M.A. van Dijk & Prof. P.A.G. van Bergeijk, EPS-2019-465-F&A, `https://repub.eur.nl/pub/115988`

Alserda, G.A.G., *Choices in Pension Management*, Promotors: Prof. S.G. van der Lecq & Dr O.W. Steenbeek, EPS-2017-432-F&A, `https://repub.eur.nl/pub/103496`

Arampatzi, E., *Subjective Well-Being in Times of Crises: Evidence on the Wider Impact of Economic Crises and Turmoil on Subjective Well-Being,* Promotors: Prof. H.R. Commandeur, Prof. F. van Oort & Dr. M.J. Burger, EPS-2018-459-S&E, `https://repub.eur.nl/pub/111830`

Arslan, A.M., *Operational Strategies for On-demand Delivery Services,* Promotors: Prof. R.A. Zuidwijk & Dr N.A. H. Agatz, EPS-2019-481-LIS, `https://repub.eur.nl/pub/126463`

Avci, E., *Surveillance of Complex Auction Markets: a Market Policy Analytics Approach*, Promotors: Prof. W. Ketter, Prof. H.W.G.M. van Heck & Prof. D.W. Bunn, EPS-2018-426-LIS, `https://repub.eur.nl/pub/106286`

Balen, T.H. van, *Challenges of Early Stage Entrepreneurs: the Roles of Vision Communication and Team Membership Change*, Promotors: Prof. J.C.M. van den Ende & Dr M. Tarakci, EPS-2019-468-LIS, https://repub.eur.nl/pub/115654

Bernoster, I., *Essays at the Intersection of Psychology, Biology, and Entrepreneurship*, Promotors: Prof. A.R. Thurik, Prof. I.H.A. Franken & Prof. P.J.F Groenen, EPS-2018-463-S&E, https://repub.eur.nl/pub/113907

Blagoeva, R.R., *The Hard Power Of Soft Power: A behavioral strategy perspective on how power, reputation, and status affect firms,* Promotors: Prof. J.J.P. Jansen & Prof. T.J.M. Mom, EPS-2020-495-S&E, https://repub.eur.nl/pub/127681

Bouman, P., *Passengers, Crowding and Complexity: Models for Passenger Oriented Public Transport*, Prof. L.G. Kroon, Prof. A. Schöbel & Prof. P.H.M. Vervest, EPS-2017-420-LIS, https://repub.eur.nl/pub/100767

Breugem, T., *Crew Planning at Netherlands Railways: Improving Fairness, Attractiveness, and Efficiency*, Promotors: Prof. D. Huisman & Dr T.A.B. Dollevoet, EPS-2020494-LIS, https://repub.eur.nl/pub/124016

Bunderen, L. van, Tug-of-War: *Why and When Teams Get Embroiled in Power Struggles*, Promotors: Prof. D.L. van Knippenberg & Dr. L. Greer, EPS-2018-446-ORG, https://repub.eur.nl/pub/105346

Burg, G.J.J. van den, *Algorithms for Multiclass Classification and Regularized Regression*, Promotors: Prof. P.J.F. Groenen & Dr. A. Alfons, EPS-2018-442-MKT, https://repub.eur.nl/pub/103929

Chammas, G., *Portfolio Concentration*, Promotor: Prof. J. Spronk, EPS-2017-410-F&E, https://repub.eur.nl/pub/94975

Chan, H.Y., *Decoding the Consumers Brain: Neural Representations of Consumer Experience*, Promotors: Prof. A. Smidts & Dr M. A.S. Boksem, EPS-2019-493-MKT, https://repub.eur.nl/pub/124931

Dalmeijer, K., *Time Window Assignment in Distribution Networks*, Promotors: Prof A.P.M. Wagelmans & Dr R. Spliet, EPS-2019-486-LIS, https://repub.eur.nl/pub/120773

Dennerlein, T., *Empowering Leadership and Employees Achievement Motivations: the Role of Self-Efficacy and Goal Orientations in the Empowering Leadership Process*, Promotors: Prof. D.L. van Knippenberg & Dr J. Dietz, EPS-2017-414-ORG, https://repub.eur.nl/pub/98438

Dolgova, E., *On Getting Along and Getting Ahead: How Personality Affects Social Network Dynamics*, Promotors: Prof. P.P.M.A.R Heugens & Prof. M.C. Schippers, EPS-2019-455-S&E, https://repub.eur.nl/pub/119150

Duijzer, L.E., *Mathematical Optimization in Vaccine Allocation,* Promotors: Prof. R. Dekker & Dr W.L. van Jaarsveld, EPS-2017-430-LIS, https://repub.eur.nl/pub/101487

Eijlers, E., *Emotional Experience and Advertising Effectiveness: on the use of EEG in marketing*, Prof. A. Smidts & Prof. M.A.S. Boksem, Eps-2019-487-MKT, https://repub.eur.nl/pub/124053

El Nayal, O.S.A.N., *Firms and the State: An Examination of Corporate Political Activity and the Business-Government Interface*, Promotor: Prof. J. van Oosterhout & Dr. M. van Essen, EPS-2018-469-S&E, https://repub.eur.nl/pub/114683

Feng, Y., *The Effectiveness of Corporate Governance Mechanisms and Leadership Structure: Impacts on strategic change and firm performance*, Promotors: Prof. F.A.J. van den Bosch, Prof. H.W. Volberda & Dr J.S. Sidhu, EPS-2017-389-S&E, https://repub.eur.nl/pub/98470

Frick, T.W., *The Implications of Advertising Personalization for Firms, Consumer, and Ad Platfroms*, Promotors: Prof. T. Li & Prof. H.W.G.M. van Heck, EPS-2018-452-LIS, https://repub.eur.nl/pub/110314

Fytraki, A.T., *Behavioral Effects in Consumer Evaluations of Recommendation Systems*, Promotors: Prof. B.G.C. Dellaert & Prof. T. Li, EPS-2018-427-MKT, https://repub.eur.nl/pub/110457

Gai, J., *Contextualized Consumers: Theories and Evidence on Consumer Ethics, Product Recommendations, and Self-Control,* Promotors: Prof. S. Puntoni & Prof. S.T.L. Sweldens, EPS-2020-498-MKT, https://repub.eur.nl/pub/127680

Ghazizadeh, P. *Empirical Studies on the Role of Financial Information in Asset and Capital Markets*, Promotors: Prof. A. de Jong & Prof. E. Peek,EPS-2019-470-F&A, https://repub.eur.nl/pub/114023

Giurge, L., *A Test of Time; A Temporal and Dynamic Approach to Power and Ethics*, Promotors: Prof. M.H. van Dijke & Prof. D. De Cremer, EPS-2017-412-ORG, https://repub.eur.nl/pub/98451

Gobena, L., *Towards Integrating Antecedents of Voluntary Tax Compliance,* Promotors: Prof. M.H. van Dijke & Dr P. Verboon, EPS-2017-436-ORG, https://repub.eur.nl/pub/103276

Groot, W.A., *Assessing Asset Pricing Anomalies*, Promotors: Prof. M.J.C.M. Verbeek & Prof. J.H. van Binsbergen, EPS-2017-437-F&A, https://repub.eur.nl/pub/103490

Hanselaar, R.M., *Raising Capital: On pricing, liquidity and incentives,* Promotors: Prof. M.A. van Dijk & Prof. P.G.J. Roosenboom, EPS-2018-429-F&A, https://repub.eur.nl/pub/113274

Harms, J. A., *Essays on the Behavioral Economics of Social Preferences and Bounded Rationality*, Prof. H.R. Commandeur & Dr K.E.H. Maas, EPS-2018-457-S&E, https://repub.eur.nl/pub/108831

Hendriks, G., *Multinational Enterprises and Limits to International Growth: Links between Domestic and Foreign Activities in a Firms Portfolio*, Promotors: Prof. P.P.M.A.R. Heugens & Dr. A.H.L Slangen, EPS-2019-464-S&E, https://repub.eur.nl/pub/114981

Hengelaar, G.A., *The Proactive Incumbent: Holy grail or hidden gem? Investigating whether the Dutch electricity sector can overcome the incumbents curse and lead the sustainability transition*, Promotors: Prof. R.J. M. van Tulder & Dr K. Dittrich, EPS-2018-438-ORG, https://repub.eur.nl/pub/102953

Jacobs, B.J.D., *Marketing Analytics for High-Dimensional Assortments,* Promotors: Prof. A.C.D. Donkers & Prof. D. Fok, EPS-2017-445-MKT, https://repub.eur.nl/pub/103497

Jia, F., *The Value of Happiness in Entrepreneurship,* Promotors: Prof. D.L. van Knippenberg & Dr Y. Zhang, EPS-2019-479-ORG, https://repub.eur.nl/pub/115990

Kahlen, M. T., *Virtual Power Plants of Electric Vehicles in Sustainable Smart Electricity Markets*, Promotors: Prof. W. Ketter & Prof. A. Gupta, EPS-2017-431-LIS, https://repub.eur.nl/pub/100844

Kampen, S. van, *The Cross-sectional and Time-series Dynamics of Corporate Finance: Empirical evidence from financially constrained firms*, Promotors: Prof. L. Norden & Prof. P.G.J. Roosenboom, EPS-2018-440-F&A, https://repub.eur.nl/pub/105245

Karali, E., *Investigating Routines and Dynamic Capabilities for Change and Innovation*, Promotors: Prof. H.W. Volberda, Prof. H.R. Commandeur & Dr J.S. Sidhu, EPS-2018-454-S&E, https://repub.eur.nl/pub/106274

Keko. E, *Essays on Innovation Generation in Incumbent Firms,* Promotors: Prof. S. Stremersch & Dr N.M.A. Camacho, EPS-2017-419-MKT, https://repub.eur.nl/pub/100841

Kerkkamp, R.B.O., *Optimisation Models for Supply Chain Coordination under Information Asymmetry*, Promotors: Prof. A.P.M. Wagelmans & Dr. W. van den Heuvel, EPS-2018-462-LIS, https://repub.eur.nl/pub/109770

Khattab, J., *Make Minorities Great Again: a Contribution to Workplace Equity by Identifying and Addressing Constraints and Privileges*, Promotors: Prof. D.L. van Knippenberg & Dr A. Nederveen Pieterse, EPS-2017-421-ORG, https://repub.eur.nl/pub/99311

Kim, T. Y., *Data-driven Warehouse Management in Global Supply Chains,* Promotors: Prof. R. Dekker & Dr C. Heij, EPS-2018-449-LIS, https://repub.eur.nl/pub/109103

Klitsie, E.J., *Strategic Renewal in Institutional Contexts: The paradox of embedded agency*, Promotors: Prof. H.W. Volberda & Dr. S. Ansari, EPS-2018-444-S&E, https://repub.eur.nl/pub/106275

Koolen, D., *Market Risks and Strategies in Power Systems Integrating Renewable Energy*, Promotors: Prof. W. Ketter & Prof. R. Huisman, EPS-2019-467-LIS, https://repub.eur.nl/pub/115655

Kong, L., *Essays on Financial Coordination*, Promotors: Prof. M.J.C.M. Verbeek, Dr. D.G.J. Bongaerts & Dr. M.A. van Achter. EPS-2019-433-F&A, https://repub.eur.nl/pub/114516

Kyosev, G.S., *Essays on Factor Investing*, Promotors: Prof. M.J.C.M. Verbeek & Dr J.J. Huij, EPS-2019-474-F&A, https://repub.eur.nl/pub/116463

Lamballais Tessensohn, T., *Optimizing the Performance of Robotic Mobile Fulfillment Systems*, Promotors: Prof. M.B.M de Koster, Prof. R. Dekker & Dr D. Roy, EPS-2019-411-LIS, https://repub.eur.nl/pub/116477

Leung, W.L., *How Technology Shapes Consumption: Implications for Identity and Judgement*, Promotors: Prof. S. Puntoni & Dr G Paolacci, EPS-2019-485-MKT, https://repub.eur.nl/pub/117432

Li, X., *Dynamic Decision Making under Supply Chain Competition*, Promotors: Prof. M.B.M de Koster, Prof. R. Dekker & Prof. R. Zuidwijk, EPS-2018-466-LIS, https://repub.eur.nl/pub/114028

Liu, N., *Behavioral Biases in Interpersonal Contexts*, Promotors: Prof. A. Baillon & Prof. H. Bleichrodt, EPS-2017-408-MKT, https://repub.eur.nl/pub/95487

Maas, A.J.J., *Organizations and Their External Context: Impressions across Time and Space*, Promotors: Prof. P.P.M.A.R Heugens & Prof. T.H. Reus, EPS-2019-478-S&E, https://repub.eur.nl/pub/116480

Maira, E., *Consumers and Producers*, Promotors: Prof. S. Puntoni & Prof. C. Fuchs, EPS-2018-439-MKT, https://repub.eur.nl/pub/104387

Mirzaei, M., *Advanced Storage and Retrieval Policies in Automated Warehouses*, Promotors: Prof. M.B.M. de Koster & Dr N. Zaerpour, EPS-2020-490-LIS, https://repub.eur.nl/pub/125975

Nair, K.P., *Strengthening Corporate Leadership Research: The relevance of biological explanations*, Promotors: Prof. J. van Oosterhout & Prof. P.P.M.A.R Heugens, EPS-2019-480-S&E, https://repub.eur.nl/pub/120023

Nullmeier, F.M.E., *Effective Contracting of Uncertain Performance Outcomes: Allocating Responsibility for Performance Outcomes to Align Goals across Supply Chain Actors*, Promotors: Prof. J.Y.F.Wynstra & Prof. E.M. van Raaij, EPS-2019-484-LIS, https://repub.eur.nl/pub/118723

Okbay, A., *Essays on Genetics and the Social Sciences*, Promotors: Prof. A.R. Thurik, Prof. Ph.D. Koellinger & Prof. P.J.F. Groenen, EPS-2017-413-S&E, https://repub.eur.nl/pub/95489

Peng, X., *Innovation, Member Sorting, and Evaluation of Agricultural Cooperatives*, Promotor: Prof. G.W.J. Hendriks, EPS-2017-409-ORG, https://repub.eur.nl/pub/94976

Petruchenya, A., *Essays on Cooperatives: Emergence, Retained Earnings, and Market Shares*, Promotors: Prof. G.W.J. Hendriks & Dr Y. Zhang, EPS-2018-447-ORG, https://repub.eur.nl/pub/105243

Plessis, C. du, *Influencers: The Role of Social Influence in Marketing*, Promotors: Prof. S. Puntoni & Prof. S.T.L.R. Sweldens, EPS-2017-425-MKT, https://repub.eur.nl/pub/103265

Pocock, M., *Status Inequalities in Business Exchange Relations in Luxury Markets*, Promotors: Prof. C.B.M. van Riel & Dr G.A.J.M. Berens, EPS-2017-346-ORG, https://repub.eur.nl/pub/98647

Pozharliev, R., *Social Neuromarketing: The Role of Social Context in Measuring Advertising Effectiveness*, Promotors: Prof. W.J.M.I. Verbeke & Prof. J.W. van Strien, EPS-2017-402-MKT, https://repub.eur.nl/pub/95528

Qian, Z., *Time-Varying Integration and Portfolio Choices in the European Capital Markets,* Promotors: Prof. W.F.C. Verschoor, Prof. R.C.J. Zwinkels & Prof. M.A. Pieterse-Bloem, EPS-2020-488-F&A, https://repub.eur.nl/pub/124984

Reh, S.G., *A Temporal Perspective on Social Comparisons in Organizations*, Promotors: Prof. S.R. Giessner, Prof. N. van Quaquebeke & Dr. C. Troster, EPS-2018-471-ORG, https://repub.eur.nl/pub/114522

Riessen, B. van, *Optimal Transportation Plans and Portfolios for Synchromodal Container Networks*, Promotors: Prof. R. Dekker & Prof. R.R. Negenborn, EPS-2018-448-LIS, https://repub.eur.nl/pub/105248

Romochkina, I.V., *When Interests Collide: Understanding and modeling interests alignment using fair pricing in the context of interorganizational information systems,* Promotors: Prof. R.A. Zuidwijk & Prof. P.J. van Baalen, EPS-2020-451-LIS, https://repub.eur.nl/pub/127244

Schie, R. J. G. van, *Planning for Retirement: Save More or Retire Later?* Promotors: Prof. B. G. C. Dellaert & Prof. A.C.D. Donkers, EOS-2017-415-MKT, https://repub.eur.nl/pub/100846

Schouten, K.I.M., *Semantics-driven Aspect-based Sentiment Analysis*, Promotors: Prof. F.M.G. de Jong, Prof. R. Dekker & Dr. F. Frasincar, EPS-2018-453-LIS, https://repub.eur.nl/pub/112161

Sihag, V., *The Effectiveness of Organizational Controls: A meta-analytic review and an investigation in NPD outsourcing*, Promotors: Prof. J.C.M. van den Ende & Dr S.A. Rijsdijk, EPS-2019-476-LIS, https://repub.eur.nl/pub/115931

Smolka, K.M., *Essays on Entrepreneurial Cognition, Institution Building and Industry Emergence*, Promotors: P.P.M.A.R. Heugens, & Prof. J.P. Cornelissen, Eps-2019-483-S&E, https://repub.eur.nl/pub/118760

Straeter, L.M., *Interpersonal Consumer Decision Making*, Promotors: Prof. S.M.J. van Osselaer & Dr I.E. de Hooge, EPS-2017-423-MKT, https://repub.eur.nl/pub/100819

Stuppy, A., *Essays on Product Quality*, Promotors: Prof. S.M.J. van Osselaer & Dr N.L. Mead. EPS-2018-461-MKT, https://repub.eur.nl/pub/111375

Subai, B., *Demographic Dissimilarity, Information Access and Individual Performance*, Promotors: Prof. D.L. van Knippenberg & Dr W.P. van Ginkel, EPS-2017-422-ORG, https://repub.eur.nl/pub/103495

Suurmond, R., *In Pursuit of Supplier Knowledge: Leveraging capabilities and dividing responsibilities in product and service contexts*, Promotors: Prof. J.Y.F Wynstra & Prof. J. Dul. EPS-2018-475-LIS, https://repub.eur.nl/pub/115138

Toxopeus, H.S., *Financing Sustainable Innovation: From a Principal-Agent to a Collective Action Perspective*, Promotors: Prof. H.R. Commandeur & Dr. K.E.H. Maas. EPS-2019-458-S&E, https://repub.eur.nl/pub/114018

Turturea, R., *Overcoming Resource Constraints: The Role of Creative Resourcing and Equity Crowdfunding in Financing Entrepreneurial Ventures*, Promotors: Prof. P.P.M.A.R Heugens, Prof. J.J.P. Jansen & Dr. I. Verheuil, EPS-2019-472-S&E, https://repub.eur.nl/pub/112859

Valboni, R., *Building Organizational (Dis-)Abilities: The impact of learning on the performance of mergers and acquisitions*,Promotors: Prof. T.H. Reus & Dr A.H.L. Slangen, EPS-2020-407-S&E, https://repub.eur.nl/pub/125226

Vandic, D., *Intelligent Information Systems for Web Product Search*, Promotors: Prof. U. Kaymak & Dr Frasincar, EPS-2017-405-LIS, https://repub.eur.nl/pub/95490

Verbeek, R.W.M., *Essays on Empirical Asset Pricing*, Promotors: Prof. M.A. van Dijk & Dr M. Szymanowska, EPS-2017-441-F&A, https://repub.eur.nl/pub/102977

Visser, T.R., *Vehicle Routing and Time Slot Management in Online Retailing*, Promotors: Prof. A.P.M. Wagelmans & Dr R. Spliet, EPS-2019-482-LIS, https://repub.eur.nl/pub/120772

Vlaming, R. de., *Linear Mixed Models in Statistical Genetics*, Prof. A.R. Thurik, Prof. P.J.F. Groenen & Prof. Ph.D. Koellinger, EPS-2017-416-S&E, https://repub.eur.nl/pub/100428

Vries, H. de, *Evidence-Based Optimization in Humanitarian Logistics*, Promotors: Prof. A.P.M. Wagelmans & Prof. J.J. van de Klundert, EPS-2017-435-LIS, https://repub.eur.nl/pub/102771

Wang, R., *Corporate Environmentalism in China*, Promotors: Prof. P.P.M.A.R Heugens & Dr F. Wijen, EPS-2017-417-S&E, https://repub.eur.nl/pub/99987

Wang, R., *Those Who Move Stock Prices,* Promotors: Prof. P. Verwijmeren & Prof. S. van Bekkum, EPS-2019-491-F&A, https://repub.eur.nl/pub/129057

Wasesa, M., *Agent-Based Inter-Organizational Systems in Advanced Logistics Operations*, Promotors: Prof. H.W.G.M van Heck, Prof. R.A. Zuidwijk & Dr A. W. Stam, EPS-2017-LIS-424, https://repub.eur.nl/pub/100527

Wessels, C., *Flexible Working Practices: How Employees Can Reap the Benefits for Engagement and Performance,* Promotors: Prof. H.W.G.M. van Heck, Prof. P.J. van Baalen & Prof. M.C. Schippers, EPS-2017-418-LIS, https://https://repub.eur.nl/pub/99312

Wiegmann, P.M., *Setting the Stage for Innovation: Balancing Diverse Interests through Standardisation*, Promotors: Prof. H.J. de Vries & Prof. K. Blind, EPS-2019-473-LIS, https://repub.eur.nl/pub/114519

Wijaya, H.R., *Praise the Lord!: Infusing Values and Emotions into Neo-Institutional Theory*, Promotors: Prof. P.P.M.A.R. Heugens & Prof. J.P. Cornelissen, EPS-2019-450-S&E, https://repub.eur.nl/pub/115973

Williams, A.N., *Make Our Planet Great Again: A Systems Perspective of Corporate Sustainability*, Promotors: Prof. G.M. Whiteman & Dr. S. Kennedy, EPS-2018-456-ORG, https://repub.eur.nl/pub/111032

Witte, C.T., *Bloody Business: Multinational Investment in an Increasingly Conflict-Afflicted World*, Promotors: Prof. H.P.G. Pennings, Prof. H.R. Commandeur & Dr M.J. Burger, EPS-2018-443-S&E, https://repub.eur.nl/pub/104027

Ye, Q.C., *Multi-Objective Optimization Methods for Allocation and Prediction*, Promotors: Prof. R. Dekker & Dr Y. Zhang, EPS-2019-460-LIS, https://repub.eur.nl/pub/116462

Yuan, Y., *The Emergence of Team Creativity: a social network perspective,* Promotors: Prof. D. L. van Knippenberg & Dr D. A. Stam, EPS-2017-434-ORG, https://repub.eur.nl/pub/100847

Zhang, Q., *Financing and Regulatory Frictions in Mergers and Acquisitions,* Promotors: Prof. P.G.J. Roosenboom & Prof. A. de Jong, EPS-2018-428-F&A, https://repub.eur.nl/pub/103871

Kaveh Azadeh was born in Tehran (Iran) on March 21, 1988. He studied Electrical Engineering at the University of Tehran in Iran. In May 2013, he received his M.Sc. degree in Industrial Engineering from the University of Central Florida (UCF) in the USA. He then spent a year as a lead researcher in the Logistics Delivery System Design Laboratory at UCF. In 2014, Kaveh joined the department of Technology and Operations Management at Rotterdam School of Management, Erasmus University, under the supervision of Professor René De Koster and Dr. Debjit Roy.

Kaveh's research interests include the design, analysis, and optimization of intra-logistic systems focusing on the stochastic modeling and the performance evaluation of various robotized order picking systems. His works have been published in Transportation Science and have been presented in several international conferences, including INFORMS Annual Meeting, European Conference on Operational Research, POMS Annual Meeting, IFORS Conference, International Conference on Computational Logistics, and International Conference on Logistics and Maritime Systems. He has also served as an ad-hoc reviewer of various journals, including Transportation Science and OR Spectrum.

ERIM

The Erasmus Research Institute of Management (ERIM) is the Research School (Onderzoekschool) in the field of management of the Erasmus University Rotterdam. The founding participants of ERIM are the Rotterdam School of Management (RSM), and the Erasmus School of Economics (ESE). ERIM was founded in 1999 and is officially accredited by the Royal Netherlands Academy of Arts and Sciences (KNAW). The research undertaken by ERIM is focused on the management of the firm in its environment, its intra- and interfirm relations, and its business processes in their interdependent connections.

The objective of ERIM is to carry out first rate research in management, and to offer an advanced doctoral programme in Research in Management. Within ERIM, over three hundred senior researchers and PhD candidates are active in the different research programmes. From a variety of academic backgrounds and expertises, the ERIM community is united in striving for excellence and working at the forefront of creating new business knowledge.

ERIM

ERIM PhD Series
Research in Management

Erasmus University Rotterdam (EUR)
Erasmus Research Institute of Management
Mandeville (T) Building
Burgemeester Oudlaan 50
3062 PA Rotterdam, The Netherlands

P.O. Box 1738
3000 DR Rotterdam, The Netherlands
T +31 10 408 1182
E info@erim.eur.nl
W www.erim.eur.nl