8-2021

Automated Warehouse Systems: A Guideline for Future Research

Wenquan Dong
University of Tennessee, Knoxville, dwenquan@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Wenquan Dong entitled "Automated Warehouse Systems: A Guideline for Future Research." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

MIngzhou Jin, Major Professor

We have read this dissertation and recommend its acceptance:

Zhenhong Lin, James Ostrowski, Bogdan Bichescu

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

AUTOMATED WAREHOUSE SYSTEMS: A GUIDELINE FOR FUTURE RESEARCH

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Wenquan Dong

August 2021

This dissertation is dedicated to my parents, Boxian Ma and Renkui Dong for their love, support, and encouragement.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Dr. Mingzhou Jin. Thanks for offering me the opportunity to join his research group and providing me with encouragement, guidance, and persistent help. I would never have been able to start and finish this rewarding journey without his support.

To my advisory committee members, Dr. James Ostrowski, Dr. Bogdan Bichescu, and Dr. Zhenhong Lin thank you for generously offering your time and providing professional guidance. We highly appreciate your inputs in this dissertation.

My sincere thanks also go to Dr. Jie Zhuang and my uncle, Dr. Renjie Dong, for their support and encouragement along this fantastic journey.

I would also like to thank my fellow graduate students: Dr. Taner Cokyasar, Huseyin Kose, Rui Li, Jeremy Hale, Dr. Tanveer Bhuiyan, and Nawei Liu. Thank you all for sharing the graduate school life at UTK. The time we spend together having fun and helping each other with the problems in research and life is greatly precious.

I also want to thank my mentors and co-workers at Oak Ridge National Laboratory, Dr. Sachin Nimbalkar, Dr. Wei Guo, and Kristina Armstrong. Thanks for their guidance and support on our collaborative project.

Finally, but most importantly, I thank my parents, Boxian Ma and Renkui Dong, and my girlfriend Rongyun Tang, for their love and unconditional support.

Thank you all so much!

ABSTRACT

This study aims to provide a comprehensive tool for the selection, design, and operation of automated warehouse systems considering multiple automated storage and retrieval system (AS/RS) options as well as different constraints and requirements from various business scenarios.

We first model the retrieval task scheduling problem in crane-based 3D AS/RS with shuttle-based depth movement mechanisms. We prove the problem is NP-hard and find an optimality condition to facilitate the development of an efficient heuristic. The heuristic demonstrates an advantage in terms of solving time and solution quality over the genetic algorithms and the other two algorithms taken from literature. Numerical experiments illustrate that when a company tends to have multiple short planning horizons with small task batches (i.e., aims to increase the responsiveness level), adding more shuttles is helpful. However, if a company has a long planning horizon with a large task batch size, having faster cranes is more efficient to reduce the makespan.

We then focus on the impacts of the number of shuttles, operational mode, storage policies, and shuttle dispatching rules on the expected cycle time of a tier-to-tier shuttle-based storage and retrieval system. The system is modeled as a discrete-time Markov Chain to derive the shuttle distribution under each scenario create the expected travel time models. Numerical experiments indicate that class-based storage is always better than the random storage policy. The best shuttle dispatching rule under each combination of the number of shuttles, operational mode, and storage policy can be quickly identified through the expected cycle time models which are simple and computation friendly.

At last, we study the warehouse design problem considering the choice, design, and operation of 2D AS/RS and 3D AS/RS in a systematic way. The warehouse design problem under consideration aims to reduce the investment while satisfying different business needs measured by the desired throughput capacity. We propose a branch-and-bound algorithm to conquer the computational challenges. With the developed algorithm, an optimal warehouse design can be obtained under different application environments, characterized by the desired throughput capacity, inventory level and demand rate of each SKU.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION AND GENERAL INFORMATION

1.1 Challenges to Warehouse Design

The prosperity of online sales provides manufacturers and retailers opportunities to directly interact with a large number of end consumers but leads to much more frequent and smaller orders with a higher variety (Boysen et al., 2019). The things that people order online have evolved from textbooks and electronics into much more categories, such as clothing, shoes, jewelry, tools, home, and appliances. Amazon alone sold 12 million products in 2018, excluding books, media, wine and services, and products sold by Amazon marketplace sellers. Traditional manufactures, such as P&G, are also exploring possible direct interaction with end customers by supplying them goods from their plants or warehouses. After more than a twenty-year development, the competition of online sales has been well beyond low prices alone and now focuses on a combination of low prices and high responsiveness, including short lead times and largely available choices (Battini et al., 2013). For instance, consumers now expect free two-day, one-day, and even same-day shipments. Warehouses or fulfillment centers, holding goods and initiating order-picking tasks, need to be more cost-efficient, flexible, and responsive to handle a large number of Stock Keeping Units (SKUs) with highly volatile demands.

Traditional warehouses with picker-to-part operations are not equipped to handle these modern challenges because they are highly labor-intensive and space consuming to satisfy a high volume of variable demands in time (Azadeh et al., 2019a; Boysen et al., 2019); and the whole world is witnessing a dramatic increase in labor and land costs (Coombs, 2018; Statista, 2021). For instance, a traditional picker-to-parts warehouse of Amazon in Chattanooga, TN has about 3,000 pickers to handle orders and occupies about 1.2 million square feet. Traditional warehouses also have the following disadvantages: 1) poor delivery performance caused by the inability to make or keep appointments for pick-up or drop off due to long unload and load waiting times, 2) high manual loading fees incurred during shipping or receiving, 3) high damage costs from parts handling, 4) ‘ship shorts’ or back charges caused by failures to reconcile product quantities with the bill of lading causes, 5) congestion in yards during peak periods preventing trailers from being rapidly serviced, and 6) high labor turnover and safety issues due to poor working conditions in some industries (e.g., cold distribution centers) (MHI, 2018).

Innovations in warehouses and material handling (MH), propelled by automation and advanced information technologies, have appeared in recent years to address all these challenges.

Automated storage and retrieval systems (AS/RS), introduced in the 1950s as an alternative to traditional warehouses and a part of advanced manufacturing systems, provide a relevant solution (MHI, 2018; Roodbergen and Vis, 2009). The AS/RS consists of racks with storage cells that can accommodate Stock Keeping Units (SKUs), one or multiple aisles between two racks, storage/retrieval (S/R) machines (e.g., cranes) which can move along the aisles to perform the storage and retrieval tasks, input/output (I/O) positions where the retrieved SKUs are dropped off and incoming SKUs are picked up by an S/R machine (Figure 1-1). The adoption rate of AS/RS has been rapidly increased in recent years due to their benefits of full-automation and high density, which lead to high space utilization (saving up to 85% space), high time efficiency (eliminating the walking that accounted for 70% of the total manual handling time), low labor request, low mispick rate, high flexibility, and high throughput capacity (Azadeh et al., 2019b; Boysen et al., 2019; Roodbergen and Vis, 2009). The high density of AS/RS also helps to significantly reduce the energy needed to heat, cool, and light warehouses (MHI, 2021a, 2021b), which satisfies the increased social awareness of corporations and governmental regulations for sustainability.

To enhance the overall responsiveness, AS/RS should be able to access order information from different customers in real-time or nearly real-time. Advanced software systems at the enterprise or supply chain level have been well developed to provide the required information. Point of Sale (POS) or Point of Purchase (POP) data can now be shared with AS/RS through Enterprise Resource Planning (ERP) systems across functional units of an enterprise or even multiple enterprises along a supply chain through reliable network technologies. The radio frequency identification (RFID) technology is one of the common commercially used real-time information collection and sharing techniques that enables warehouse managers to closely monitor material flows (Poon et al., 2011). Besides software, the development of technologies also enables the innovation of hardware systems, such as cranes, lifts, and shuttles, so that AS/RS became more applicable to more industries and under more operational settings regarding both technical and economic feasibility.

Traditionally, storage in racks may be single-deep or double-deep (2D) so that only one or two SKUs can be stored. Even though 2D AS/RS has significantly improved land utilization, the aisles between racks can consume about 35% of land space (Stadtler, 1996). Moreover, 2D AS/RS have relatively long travel times to store and retrieve unit loads. Therefore, since their

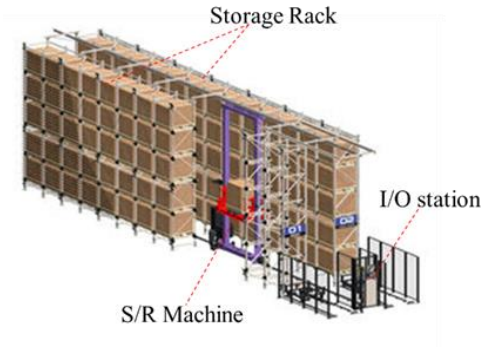


Figure 1-1. An Overview of Traditional AS/RS (Bastian Solutions, 2010)

introduction in the 1950s, AS/RSs have gone through many alterations to fit different business settings and to adopt new hardware and software advances. Therefore, numerous AS/RSs are currently available for companies to consider (Figure 1-2). In terms of the depth of racks used to accommodate SKUs, the AS/RS can be defined as 2D AS/RS (traditional AS/RS) and 3D AS/RS (also known as compact/high-density storage systems). The storage in racks of a 2D AS/RS might occur single deep or double deep, while compact storage systems (3D) have racks with deep lanes that can be used to store multiple SKUs. In terms of S/R machines, AS/RS can be classified into carousel-based, lift/shuttle-based, and crane-based systems. If we consider some specific operational mechanisms, there are even more variations, such as Aisle-captive AS/RS and Aisle-changing AS/RS (Roodbergen and Vis, 2009). Emerging technologies, such as robot-based compact storage and retrieval (RCSR) Systems (Zou et al., 2017), which can provide very high storage density without allocating space for aisles and high flexibility due to the expandability of a robot fleet, can further expand the advantages of AS/RS.

Different industries and companies have their own needs, requirements, and constraints, which means no single AS/RS is appropriate for all purposes. For instance, compared to a 2D system, a 3D AS/RS can further improve land space utilization and energy efficiency. However, for industries/companies that need to handle a large number of orders with different sizes and a high turnover rate, a 2D system may be a better choice (Boysen et al., 2019). In addition, the investment cost for hardware and software is high and usually not reversible. A single AS/RS rack with a crane may cost over $700,000 (Roodbergen and Vis, 2009). Therefore, the rapid development of a large variety of AS/RS brings huge opportunities for their applications in many situations but also poses a challenge for companies to choose the right system and decide the right configuration for their specific needs. Those needs are highly diversified across companies. Until now, there is no guidance, models, and tools for this critical decision-making in both the academic community and industry, which has become a barrier for businesses adopting AS/RS systems.

1.2 Literature Review

Even though the selection and design of the appropriate AS/RS affects the overall warehouse performance, existing studies are preliminary and only considered AS/RS selection by assuming

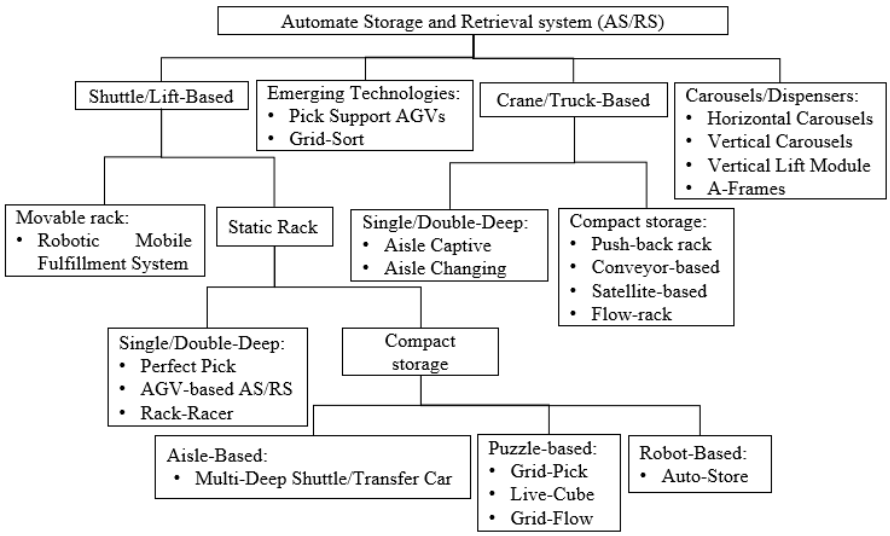


Figure 1-2. Categories of AS/RS (Azadeh et al., 2019)

the physical design, operation policies, and performance of the candidate AS/RS systems are known in advance (Azadeh et al., 2019a; Jaghbeer et al., 2020). Shen et al. (2010) focused on a warehouse design problem considering manual and semi-automated warehouse systems as potential options to minimize pickers' travel time. However, they assumed all the SKUs are the same and did not have to consider the SKUs-to-system assignment. In addition, Pazour & Meller (2014) is the first that modeled the warehouse design problem considering multiple order fulfillment technologies (e.g., AS/RS and manual warehouse system). However, they assumed the physical design, operation policies, and the performance (i.e., throughput capacity) of each technology are known in advance, and only considered the technology selection and SKU-to-system assignment. Roodbergen et al. (2015) considered a warehouse design problem by allowing the mixed-use of aisle-captive and aisle-to-aisle 2D AS/RS via simulation but assumed all SKUs are identical.

Except for the limited attention paid to the selection of AS/RS, a few studies have been done to compare the automated vehicle storage and retrieval system (AVS/RS) and the conventional 2D AS/RS. For instance, Heragu et al. (2011) modeled AVS/RS and the conventional 2D AS/RS as open queuing networks and compared the performance of these two systems with a set of pre-determined physical designs and operation policies. Later, Ekren & Heragu (2012) compared the performance of AVS/RS with the aisle-to-aisle 2D AS/RS, which allows a crane to serve different racks, in terms of different performance metrics by simulating these two systems with different configurations. However, similar to Heragu et al. (2011), this study assumed the configurations of the two systems are known in advance. The tier-to-tier AVS/RS and tier-captive AVS/RS are also compared by Küçükyaşar et al. (2021) and the similar assumptions with the previous system comparison studies are adopted, and therefore, limited its impacts on AS/RS selection and design.

Current researches on AS/RS mainly focuses on micro-level design and operations for an AS/RS system, investigating how to improve its operational efficiency measured using different metrics such as throughput capacity and cycle time for a given set of tasks via physical design and system operation (Azadeh et al., 2019a; Boysen et al., 2019; Davarzani and Norrman, 2015; De Koster et al., 2017). Roughly, current studies related to AS/RS can be classified into three categories: system performance analysis; long-term decision problems (physical design), and short-term decision problems (operation policies); studies within both of these categories mainly consider a single-rack system.

System analysis articles focus on estimating systems' performance in terms of one or more performance metrics (e.g., average cycle time, energy consumption, and transaction waiting time). The average cycle time for finishing one transaction, which indicates the systems throughput capacity, is one of the commonly used system performance metrics. Cycle time models, which are usually simple and computationally friendly, are commonly used to give the closed-form formulation for estimating the average cycle time (D'Antonio et al., 2018; D'Antonio and Chiabert, 2019; Sari et al., 2005). Queueing models and simulation models are used to measure the other performance metrics (e.g., transactions' waiting time) in an AS/RS (Cai et al., 2014; Roodbergen et al., 2015; Schenone et al., 2020). Long-term decision problems mainly consider one of the following problems: the dimension of racks (Azadeh et al., 2019b; Xu et al., 2019b; Yang et al., 2015a; Yu and De Koster, 2009a), number and locations of I/O positions (Xu et al., 2019a), number and characteristics (e.g., speed profile) of equipment (Ha and Chae, 2018, 2019). Studies about short-term decision problems usually consider task sequencing (Dong et al., 2021; Fang and Tang, 2014; Yu and De Koster, 2012), task batching (Li et al., 2017), and storage assignment (Schenone et al., 2020; Yu et al., 2015). Among those problems, the most frequently studied ones are rack dimension problem, storage assignment problem, and task sequencing problems under some specific assumptions, such as a constant equipment speed, a single-command (SC) cycle or double-command (DC) cycle for crane operations, etc. These studies can provide some insights, such as the advantages of turnover-based storage assignment policy over random storage assignment (Yu and De Koster, 2009a) and the DC operations has better performance than the SC operations if both storage and retrieval tasks are considered in a planning horizon (De Koster et al., 2008). However, all of the aforementioned studies focused on the design and operation of AS/RS with a single-rack assumption, which limited the insights for facilitating warehouse design in practice. Furthermore, some of the AS/RS systems that were introduced recently have not received any attention. Moreover, advanced AS/RSs such as crane-based 3D AS/RS and puzzle-based AS/RS are very complex and models used to study them are often computationally expensive (Dong et al., 2021). For practitioners, it is difficult and time-consuming to create and solve those complex mathematical models for every possible configuration of every AS/RS under consideration and choose the best one.

A few studies have been done considering the design of 2D AS/RS which has multi-racks. Bozer & White (1990) considered the warehouse design problem with 2D AS/RS to determine the

number and dimensions of racks in a warehouse to minimize the hardware cost while meeting a given throughput capacity. However, they assumed identical SKUs and did not consider the impact of demand rate and inventory level of different SKU types in the warehouse. Manzini et al. (2006) and Rao & Adil (2013) also considered the warehouse design problem with 2D AS/RS by determining the number of aisles to minimize the average transaction time. However, again, they assumed identical SKUs. Mital et al. (2015) studied the warehouse design problem with 2D AS/RS by selecting the design from a set of predefined configurations to minimize the cost and system risk.

In summary, AS/RS-related researches mainly considered the design and operations of AS/RS systems with the assumption of a single-rack system. Even though a few studies have been done considering the AS/RS design by allowing multi-racks, or the comparison and selection of a few AS/RS systems (i.e., AVS/RS and 2D AS/RS), these studies either ignore the differences between different SKUs (e.g., inventory level and demand rate) or assume predefined physical design and operations policies, which is far away from practice. This dissertation aims to fill this gap by considering the warehouse design problem with multiple AS/RS options under different business needs.

1.3 Document Organization

To facilitate the warehouse design by comparing, selecting, and designing multiple AS/RS options, the design, operation, and performance of each AS/RS should be completely studied. Therefore, this dissertation started with decisions at the operational level and then moved to design and choice decisions.

Chapter 2 considered how to sequence a group of retrieval requests in a crane-based 3D AS/RS with shuttle-based DMMs to minimize the makespan (i.e., total cycle time). A good sequence can significantly reduce the makespan for finishing a given group of retrieval tasks. However, the task scheduling problem has never been studied for crane-based 3D AS/RS with shuttle-based DMMs, which has become increasingly popular in practice. A mixed-integer programing model was developed to represent the problem, and the problem was proven to be NP-hard. An optimality condition of the scheduling problem was proved and based on what, a heuristic was developed. A novel chromosome structure was also proposed to apply the Genetic Algorithm to solve the problem quickly. The numerical experiments indicate the advantages of the proposed heuristic

over the genetic algorithm and the other two algorithms found from literature in terms of solving time and solution quality. Numerical results also revealed insights for improving 3D AS/RS productivity.

The performance of a tier-to-tier shuttle-based storage and retrieval system (SBS/RS) or 3D AS/RS with shuttle-based DMMs is affected by both operation mode (single-command (SC) cycles and dual-command (DC) cycles), storage policy, and shuttle dispatching rule. Chapter 3 explored the impact of both SC and DC operations, random and class-based storage policies, and three shuttle dispatching rules (e.g., random, distance-based, and demand-rate based) on the performance of tier-to-tier SBS/RS. Modeling the system as a discrete-time Markov Chain, this chapter derived the shuttle distribution under each policy and operations mode combination, and further developed the expected cycle time models, which were validated by simulation. Numerical experiments showed significant impacts of the policy combination on the expected cycle time. The best storage and shuttle dispatching policy under different scenarios were also identified.

Chapter 4 studied the warehouse design problem (characterized by different distributions of inventory levels and demand rates of different SKU types) by considering 2D AS/RS and 3D AS/RS as options to reduce the warehouse investment while maintaining a certain level of throughput capacity. The warehouse design problem was first modeled as mixed-integer nonlinear programming and then was converted to mixed-integer programming based on optimality conditions. A branch-and-bound algorithm was developed for the computational challenges and was modified to further reduce the solving time. Numerical experiments showed the impacts of the cost parameters, and the distribution of the inventory levels and demand rates of different SKU types on warehouse design.

At last, Chapter 5 summarized this work and discussed the further research directions.

CHAPTER TWO RETRIEVAL SCHEDULING IN CRANE-BASED 3D AS/RS WITH SHUTTLES-BASED DMMS

2.1 Abstract

Retrieval task scheduling has been extensively studied for 2D automated retrieval and storage systems (AS/RS). A good schedule can significantly reduce the makespan for finishing a given group of retrieval tasks. However, the task scheduling problem has never been studied for crane-based 3D AS/RS with shuttle-based depth movement mechanisms (DMMs), which has become increasingly popular in practice. This chapter considered how to schedule a group of retrieval requests in a crane-based 3D AS/RS with shuttle-based DMMs to minimize the makespan. A mixed-integer programing model was developed to represent the problem, and the problem was proven to be NP-hard. Four heuristics were investigated for their computational performance. First-Come-First-Serve is the current practice while the Percentage Priority to Shuttle Reallocation with the Shortest Leg (PPS-SL) rule was developed based on the existing rule for scheduling storage and retrieval tasks in 3D AS/RS with conveyor-based DMMs. The Genetic Algorithm, which is popular for 2D systems, was adapted to deal with the 3D system. The Lowest-Waiting-Time-First heuristic was proposed based on the optimality condition of the scheduling problem and was demonstrated to outperform the other three algorithms in terms of solution quality and computational time. Further numerical results revealed insights for improving 3D AS/RS productivity. When the number of retrieval tasks is small (e.g., when a short planning horizon is adopted for high responsiveness), having more shuttles can improve the system performance. When there are many tasks to schedule, for example, in a situation with a long planning horizon, using a higher crane speed rather than adding more shuttles can improve system efficiency more.

2.2 Introduction

Because of growing online sales and the popularity of continuous inventory replenishment, companies are fulfilling highly varied demands, are in small batches, and require quick responses from a great number of customers (Boysen et al., 2015; Gaku and Takakuwa, 2018; Tutam and White, 2019). The automated storage and retrieval system (AS/RS), which was introduced in the 1950s, is an important technology for improving productivity in warehouses (Chang and Egbelu, 1997; Man et al., 2019). AS/RS is a computer-controlled automated material handling system that can be used to store incoming products and materials and retrieve stock-keeping units (SKUs) without direct handling by labor (Roodbergen and Vis, 2009). The main AS/RS components are

racks, storage/retrieval (S/R) machines (crane), and input/output (I/O) stations (Figure 1-1). Racks are structures with storage cells that accommodate SKUs. Cranes can move along aisles between two racks to drop off or pick up SKUs. An I/O point is where retrieved SKUs are dropped off and where incoming SKUs are picked up by a crane. AS/RS has been widely used because of the advantages of full automation and space savings (Boysen and Stephan, 2016; MHI, 2018). Traditionally, a storage spot in a rack may be single deep or double deep (2D) and hold only one or two SKUs. Even though the 2D AS/RS has significantly improved land use, the aisles between racks can consume 35% of land space (Stadtler, 1996), which also increases travel times to store and retrieve SKUs.

Improved use of space has been achieved by the 3D AS/RS (also called multi-deep, compact, or super high-density storage systems) that consists of racks with deep lanes that can store more than two SKUs. A 3D AS/RS with cranes as S/R machines is called the crane-based 3D AS/RS (Figure 2-1). In that system, each SKU can be accessed by the coordination of a crane and depth movement mechanisms (DMMs). The crane takes care of the movement in horizontal and vertical directions (*x, y* directions in Figure 2-1) and DMMs move SKUs along each lane (*z*-direction). Compared with a 2D system, a 3D system can produce shorter lead times, lower energy consumption, and less space consumption (Azadeh et al., 2019a). The crane-based 3D AS/RS has become increasingly popular, and its applications can be found in many studies (Yu and De Koster, 2009b, 2009a) and under various commercial settings, such as food manufacturing and E-commerce fulfillment centers. This study examines the crane-based 3D AS/RS with shuttle-based DMMs. In such a system, shuttles can travel along lanes to transfer SKUs. Cranes move not only SKUs to I/O points but also empty shuttles across lanes for two reasons: 1) The number of shuttles is usually smaller than the number of lanes because automated shuttles are usually expensive; 2) Reallocating shuttles increases the system flexibility and reliability because the system operation will not be halted due to shuttle failure or maintenance.

The performance of AS/RS relies on task scheduling (Li et al., 2017). In practice, First-Come-First-Serve (FCFS) is commonly used for scheduling tasks. However, rescheduling tasks can significantly improve system performance (Dooly and Lee, 2008; Lee, 1997). In a 3D AS/RS, each task is finished by the cooperation of a crane and a DMM, which requires more sophisticated scheduling.

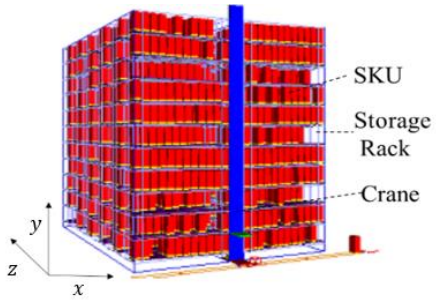


Figure 2-1. An Overview of Crane-based 3D AS/RS (Yu and De Koster, 2012)

In practice, tasks in an AS/RS can be conducted in two modes:

- Single Cycle (SC): A crane executes one retrieval or storage task in one operational cycle. For a retrieval task, the crane departs from the I/O point, travels to the target lane/storage position, loads the SKU, and travels back to the I/O point. The storage operation is similar.
- Combined Cycle: A crane executes one retrieval and one storage task in one operational cycle. The crane picks up an SKU from the I/O point, moves to an open location on the rack and unloads the SKU, travels to another target lane, picks up an SKU, and then travels back to the I/O point. In literature, the combined cycle is also called dual-command Cycle (DC). In this study, these two terms are used interchangeably.

The task scheduling problem in a 2D system with the DC mode was shown to be NP-hard (Han et al., 1987). The operations do not become easier even though we only focus on retrieval tasks with the SC mode in a 3D AS/RS for the following three reasons: 1) The number of moving elements in a 3D AS/RS is greater than in a 2D AS/RS. For instance, the 3D system considered in this study has 240 lanes, which means the crane may need to cooperate with up to 240 DMMs. 2) For the system with shuttle-based DMMs, we need to reassign shuttles across lanes. 3) The problem can be much more difficult if there are different types of SKUs in one lane. One SKU might block the way of the requested SKU. In that situation, DMMs and cranes need to work together to move the SKU that blocks the way of the requested SKU to another lane (also called reshuffle).

This study was motivated by a collaboration with a major system designer and manufacturer of AS/RS systems. They have delivered and operated crane-based 3D AS/RS with shuttle-based DMMs for various clients, including cigarette distributors, pharmaceutical distributors, E-commerce companies selling fashion products, online retailers of cosmetic products, etc. All those products have a large demand volume and small sizes. Responsive fulfillment of customer demands is their key competitive strategy. Inventory replenishment is not as urgent as retrieval tasks so they adopt the SC mode and execute retrieval tasks and storage tasks in different time windows to avoid the complexity of scheduling both retrieval and replenishment tasks simultaneously. Therefore, this study only considers the retrieval task scheduling problem with the SC mode in a crane-based 3D AS/RS with shuttled-based DMMs, especially for growing online sales businesses. This study proposes a mixed-integer programming (MIP) model

intending to minimize the makespan of completing a given set of retrieval tasks. It is worth to note that the problem can be modeled more straightforwardly by having many more binary variables. We carefully created an MIP model that has fewer binary variables and therefore less computational burden. Section 2.4 compares the two modeling methods. As requests arrive continuously in practice, the system has to update the task schedule frequently, which demands a short computational time for the scheduling problem. Therefore, the study proposed the Lowest-Waiting-Time-First solution method that incorporates the optimality condition of the scheduling problem. The proposed heuristic is compared with the Genetic Algorithm, First-Come-First-Serve, and Shortest Leg, which was adapted from literature for 3D systems with conveyor-based DMMs, to demonstrate its advantage. Furthermore, as the number of shuttles ranges from one to the number of lanes, this study provides insights on the desired number of shuttles to balance system efficiency and cost through extensive numerical experiments that will use data from the abovementioned AS/RS OEM. At last, this study will also contribute to the study on retrieval scheduling problems in tier-to-tier SBS/RS, which has a similar operational mechanism to crane-based 3D AS/RS with shuttle-based DMMs.

The remainder of this chapter is organized as follows. Section 2.3 reviews previous studies related to the task scheduling problem in AS/RS. Section 2.4 gives an overview of the operational mechanism of the system and an MIP model to describe the problem. In Section 2.5, four solution methods are presented to solve the problem in a reasonable amount of time. Section 2.6 presents the numerical results to compare the efficiency of the proposed solution methods and obtain managerial insights to improve the system. Finally, section 2.7 concludes the paper and discusses the future work.

2.3 Literature Review

The task scheduling problem in 2D AS/RS has been thoroughly studied (Azadeh et al., 2019a; Boysen and Stephan, 2016; Gagliardi et al., 2012; Roodbergen and Vis, 2009). As the crane is the only moving element in a 2D system, the modeling method used for the scheduling problem in a 2D system does not fit this study. To model the task scheduling problem in a 3D AS/RS system, we must formulate the collaboration between a crane and DMMs.

The scheduling problem in a 2D AS/RS that operates in the DC mode with multi-open locations was proven to be NP-hard (Han et al., 1987; van den BERG and (NOUD) Gademann, 1999).

Various dispatch rules, such as FCFS, Nearest Neighbor, Shortest-Leg, Shortest DC cycle have been developed (Man et al., 2019; Yu and De Koster, 2012). These methods are easy to implement but often converge to local optimums. To search a larger feasible region for possible global optimums, metaheuristics such as the Ant Colony Algorithm (Li et al., 2017), GA (Noorul Haq et al., 2003; Wu et al., 2013), Tabu Search (Bessenouci et al., 2012; Yang et al., 2015b) and Simulated Annealing (Bessenouci et al., 2012) have been used to solve scheduling problems in 2D AS/RS. GA is commonly used and can usually provide a good solution in a reasonable amount of time (Chetty and Reddy, 2003; Noorul Haq et al., 2003; Popović et al., 2014). Some studies on 3D AS/RS have been done, but they focus on storage assignments (Yu and De Koster, 2009b, 2009a) and physical design problems (De Koster et al., 2008; Yang et al., 2015a, 2017).

Yu and De Koster (2012) were the first who considered the task scheduling problem in a 3D system. They focused on a 3D system with conveyor-based DMMs in the DC mode. The scheduling problem for a block of retrieval and storage tasks was expressed mathematically and proven to be NP-hard. However, they stated that their mode was conceptual with strong nonlinear constraints and was impossible to solve, even for small case problems. They modified the shortest-leg heuristic by giving priority to preposition retrieval tasks (PPR-SL) and illustrated the advantage of the modified algorithm. Moreover, with conveyor-based DMMs, they did not have to consider the reallocation of DMMs. However, the reallocation of DMMs (i.e., shuttles) makes our study completely different from the task scheduling problem in a 3D system with conveyor-based DMMs. Even though we only consider retrieval cycles, the scheduling problem is more complicated than Yu and De Koster (2012). A detailed explanation can be found in Appendix A. Moreover, in contrast to our study, they ignored the acceleration and deceleration of the S/R machine. Ignoring this could result in a deviation from real-world cases, and the deviation increased when there was higher skewness of the inventory distribution and more storage classes (Wen et al., 2001). To the best of our knowledge, Zaerpour et al. (2015) were the first to study crane-based 3D AS/RS with shuttle-based DMMs. Different from our study, they focused on the storage assignment problem. In addition, in their study, instead of multiple shuttles, there is only one shuttle for the crane. The crane has to wait while the shuttle travels along a lane.

Autonomous vehicle storage and retrieval system (AVS/RS), another AS/RS type, has a similar operational mechanism to crane-based 3D AS/RS with shuttle-based DMMs. An AVS/RS (including 2D and 3D) adopts lifts and shuttles for vertical movement and horizontal movement,

respectively. AVS/RS can be categorized into tier-captive and tier-to-tier AVS/RS. In a tier-captive system, each tier has its own shuttle. In a tier-to-tier AVS/RS, the system typically has fewer shuttles than tiers, and lifts reallocate shuttles across tiers. AVS/RS has received lots of academic attention, which mainly focus on system performance analysis and configuration design through travel time models (Fukunari and Malmborg, 2008; Ha and Chae, 2019; Lerher et al., 2015b; Lerher, 2016), queueing network (Cai et al., 2014; Ekren et al., 2013; Tappia et al., 2017), and simulation (Ekren and Heragu, 2010, 2012). However, studies on task scheduling in AVS/RS are very limited. Carlo and Vis (2012) studied the scheduling problem of a tier-captive AVS/RS with multiple lifts. However, they assumed the tasks schedule was given in advance and only considered pairing shuttles/tasks with lifts. Wang et al. (2015) considered the retrieval tasks scheduling problem in a tier-captive AVS/RS with one lift. Later, they extended the study by considering a tier-to-tier system (Xu et al., 2019b). However, they assumed that empty shuttles were always pooled at the ground level, and when conducting a task, a shuttle had to ride on a lift to travel to the target tier and go back to the ground level with the requested SKU. Therefore, shuttles do not travel directly across tiers. Recently, tier-to-tier AVS/RS allowing shuttles to move across tiers, which is also called as tier-to-tier SBS/RS, starts to get more attention (Ha and Chae, 2018, 2019; Wu et al., 2020). Cao and Zhang (2017) were the first considering scheduling problem in a 3D tier-to-tier SBS/RS. They considered at most two shuttles, and the nearest neighbor rule was applied whenever a shuttle reallocation was needed. However, when more shuttles are added into the system, the nearest neighbor rule may not be the optimal. Therefore, our study, which allows more than two shuttles in a system, can also contribute to retrieval scheduling problems in tire-to-tier SBS/RS.

As discussed above, even though many studies have examined the task scheduling problem in AS/RS, there is no research solving the scheduling problem in a 3D system with shuttle-based DMMs even though that system is used widely due to its capability of handling a large variety of orders with high frequency—a trend driven by increasing online sales. Our study is the first to look at the scheduling problem in the 3D system with shuttle-based DMMs. We propose an MIP model to express the problem in the next section, which is proven NP-hard. The Lowest-Waiting-Time-First (LW) heuristic based on the optimality condition was developed to provide quality solutions in a short computational time. The heuristic was compared with the Genetic Algorithm

(GA), First-Come-First-Serve (FCFS), and Shortest Leg to demonstrate its efficiency in terms of solution quality and solving time.

2.4 Problem Description and Formulation

This study considers the retrieval tasks of a rack in a crane-based 3D AS/RS with shuttle-based DMMs that allows fewer shuttles than lanes. The crane operates in single cycles (SC) and handles two types of tasks, retrieval and shuttle reallocation. To accomplish a retrieval task, the shuttle in the lane of the requested SKU, if there is one, will move the SKU to the end of the lane, where the crane picks up the SKU and transfers it to the I/O point. When conducting a shuttle reallocation task, the crane travels to a lane, picks up the shuttle located there, and moves it to another lane. As there are multiple shuttles on one rack and they can work independently, the crane may execute other tasks during one shuttle's operational cycle. The crane is assumed to remain at the position where it finished its last job before performing a new job. SKUs on one lane are assumed to be the same so there is no need to consider a reshuffle.

Let N^0 and N^1 represent the set of lanes with a shuttle and the set of lanes without a shuttle but having retrieval tasks at the beginning, respectively. Please note that we do not need to consider the lanes without any shuttles and retrieval tasks in our scheduling problem. However, it is worth to note that, in practice, a system may have more lanes than the SKU types and N^0 should be decided based on a lane-to-task assignment problem. The lane-to-task assignment problem can be modeled based on the MIP model proposed for the scheduling problem (2.1-2.21) and can be found in Appendix B. The cardinalities of these two sets (i.e., $|N^0|$ and $|N^1|$) can represent the number of shuttles and number of shuttle reallocation tasks, respectively. Let N denote the set of lanes considered in a planning horizon (i.e., $N = N^0 \cup N^1$), with i and j denoting indices. Q_i, $Q' = \max_{i \in N} Q_i + 1$, and $M = |N^1| + \sum_{i \in N} Q_i$ are the number of retrieval tasks from lane i, the largest number of possible tasks at a lane over all lanes, and the number of total tasks (retrieval tasks plus shuttles reallocation tasks). Here, $\underset{i \in N}{\text{Max}}\, Q_i$ is the largest number of retrieval tasks at a lane over all lanes and one is added to get Q' due to the possibility of moving a shuttle from this lane.

Time parameters are introduced as the performance metrics of the system. We define p_{iq}^1 as the time for a shuttle finishing the q^{th} task in lane i where $q \in \{1, \dots, Q_i + 1\}$. When $q < Q_i + 1$, p_{iq}^1

represent the time for the shuttle to travel to the q^{th} task in lane i from the lane end, load the SKU, and travel back to the lane end. The $(Q_i + 1)^{\text{th}}$ task for lane i happens if the shuttle on lane i needs to be transferred to another lane. When retrieval tasks from this lane are all finished, the shuttle just needs to wait at the lane end for being picked up by the crane so that $p^1_{i,Q_i+1} = 0$. The time for the crane traveling from lane i to lane j is defined as p^2_{ij}. We also have p^3_{ij} represent the time for the crane traveling from lane i to the I/O point, unloading an SKU, and then traveling to lane j for the next task. Parameters p^4_i and $P_{i_0,i}$ denote the time for the crane traveling from lane i to the I/O point and unloading a unit, and the time for the crane traveling from the original position to lane i, respectively. The planning horizon T serves as a big number to facilitate the formulation in (2.15-2.20).

Various binary variables are defined for scheduling decisions. If the m^{th} single-command cycle (SC) task is for retrieving an SKU or a shuttle from lane i, $y_{mi} = 1$; otherwise, $y_{mi} = 0$. We also define $z_{mq} = 1$ if the m^{th} SC task is for conducting the q^{th} task at its associated lane. If the m^{th} SC task is to move a shuttle, $\theta_m = 1$; otherwise, $\theta_m = 0$. To express the reallocation of shuttles, we define x_{ij} to denote whether a shuttle is moved from lane i to lane j ($x_{ij} = 1$) or not ($x_{ij} = 0$). In addition, t_m and r_{iq} represent the moment when the crane picks up the SKU/shuttle scheduled as the m^{th} SC task and the moment when the q^{th} task in lane i is ready for the crane to pick up. l_i and u_i, representing the task indices of the first and last tasks associated with lane i, are introduced for avoiding sub-tours of shuttles and making sure that the crane only visits a lane that has a shuttle. The mixed-integer program (MIP) for the scheduling problem is provided as follows,

$$Min \quad t_M + \sum_{i \in N} p^4_i y_{Mi} \tag{2.1}$$

$$s.t. \quad \sum_{i \in N/\{j\}} x_{ij} = 1, \qquad j \in N^1, \tag{2.2}$$

$$\sum_{j \in N^1} x_{ij} \leq 1, \qquad i \in N, \tag{2.3}$$

$$\sum_{m=1}^{M} y_{mi} = Q_i + \sum_{j \in N^1} x_{ij}, \qquad i \in N, \tag{2.4}$$

$$\sum_{i \in N} y_{mi} = 1, \qquad m \in \{1, \dots, M\}, \tag{2.5}$$

$$u_i \geq m y_{mi}, \qquad i \in N, m \in \{1, \dots, M\}, \tag{2.6}$$

$$l_i \leq m y_{mi} + M(1 - y_{mi}), \qquad i \in N, m \in \{1, \dots, M\}, \tag{2.7}$$

$$u_i \leq l_j + M\left(1 - x_{ij}\right) \qquad i \in N, j \in N^1, \tag{2.8}$$

$$\sum_{q=1}^{Q_i+1} q z_{mq} \leq \sum_{k=1}^{m} y_{ki} + (Q_i + 1)(1 - y_{mi}), \qquad i \in N, m \in \{1, \dots, M\}, \tag{2.9}$$

$$\sum_{q=1}^{Q_i+1} q z_{mq} \geq \sum_{k=1}^{m} y_{ki} - (Q_i + 1)(1 - y_{mi}), \qquad i \in N, m \in \{1, \dots, M\}, \tag{2.10}$$

$$\sum_{q=1}^{Q'} z_{mq} = 1, \qquad m \in \{1, \dots, M\}, \tag{2.11}$$

$$Q'(1 - \theta_m) \geq \sum_{i \in N} (Q_i + 1) y_{mi} - \sum_{q=1}^{Q'} q z_{mq}, \qquad m \in \{1, \dots, M\}, \tag{2.12}$$

$$\theta_m \geq \sum_{q=1}^{Q'} q z_{mq} - \sum_{i \in N} Q_i y_{mi}, \qquad m \in \{1, \dots, M\}, \tag{2.13}$$

$$t_1 \geq \sum_{i \in N^0} P_{i_0 i} y_{1i}, \tag{2.14}$$

$$t_m \geq t_{m-1} + p_{ij}^3 - T\left(2 - y_{m-1,i} - y_{mj} + \theta_{m-1}\right), \quad i, j \in N, m \in \{2, \dots, M\}, \tag{2.15}$$

$$t_m \geq t_{m-1} + \sum_{k \in N} \left(p_{ik}^2 + p_{kj}^2\right) x_{ik} - T\left(3 - y_{m-1,i} - y_{mj} - \theta_{m-1}\right), \quad i, j \in N, m \in \{2, \dots, M\}, \tag{2.16}$$

$$r_{i1} \geq p_{i1}^1 - T(2 - z_{m1} - y_{mi} + \theta_m), \quad i \in N^0, m \in \{1, \dots, M\}, \tag{2.17}$$

$$r_{i1} \geq t_m + p_{ij}^2 + p_{i1}^1 - T\left(3 - z_{m,Q_j+1} - y_{mj} - \theta_m\right), \quad i \in N^1, j \in N, m \in \{1, \dots, M\}, \tag{2.18}$$

$$r_{iq} \geq t_m + p_{iq}^1 - T\left(2 - z_{m,q-1} - y_{mi}\right), \quad i \in N, q \in \{2, \dots, Q_i + 1\}, m \in \{1, \dots, M\}, \tag{2.19}$$

$$t_m \geq r_{iq} - T\left(2 - y_{mi} - z_{mq}\right), \quad i \in N, q \in \{1, \dots, Q_i + 1\}, m \in \{1, \dots, M\}, \tag{2.20}$$

$$t_m, r_{iq}, u_i, l_i \geq 0; y_{mi}, x_{ij}, \theta_m, z_{mq} \in \{0,1\}. \tag{2.21}$$

The objective function (2.1) minimizes the total cycle time for the crane to complete a given batch of retrieval tasks. Constraint set (2.2) makes sure that a lane in N^1 receives a shuttle exactly once. Constraint set (2.3) guarantees that the crane moves a shuttle out of a lane in N^0 up to once. Constraint set (2.4) assures that the number of SC tasks related to lane i is the number of retrieval tasks at lane i plus one shuttle reallocation task if there is any. Constraint set (2.5) ensures that the crane only performs one retrieval or shuttle reallocation task during one SC. Constraint sets (2.6) through (2.8) make sure that the crane can only serve a lane that has a shuttle and avoid the sub-tour problem. In other words, when a shuttle is moved from lane i to lane j, all tasks in lane i should be finished before the first task in lane j starts. Constraint sets (2.9) through (2.11) force z_{mq} to be one if the q^{th} task on a lane is scheduled as the m^{th} SC task. Constraint sets (2.12) and

(2.13) work together to guarantee θ_m to be one if the m^{th} SC task is for moving a shuttle (i.e., it is the $(Q_i + 1)^{\text{th}}$ for lane i, which is the lane for the m^{th} SC task). Constraint sets (2.14) through (2.16) make sure that the m^{th} SC task cannot start until the $(m-1)^{\text{th}}$ SC task is completed. Specifically, constraint set (2.14) is for the case of the first SC task; (2.15) is for the case when the $(m-1)^{\text{th}}$ SC task is to retrieve an SKU; and (2.16) is for the case when the $(m-1)^{\text{th}}$ SC task is to move a shuttle to lane j. Constraint sets (2.17) through (2.20) are used to obtain the time when the shuttle with the SKU for the q^{th} task at lane i is ready for pick up by the crane. Constraint set (2.17) is used for the first retrieval task on lanes in N^0; (2.18) is used for the first retrieval task on lanes from N^1; and (2.19) makes sure that the shuttle cannot move for the q^{th} SKU before the crane picks up the $(q-1)^{\text{th}}$ SKU from that lane. Constraint set (2.20) calculates the moment when the crane picks up the SKU or shuttle in its m^{th} cycle.

The proposed MIP carefully uses y_{mi} and z_{mq} together to represent the crane's route and task sequence. Multiple constraint sets are created to connect y_{mi} and z_{mq}. Even though the problem can be formulated more straightforwardly by combining y_{mi} and z_{mq} to a set of three-dimensional variables, the model with three-dimensional variables will have much more binary variables, and therefore, has a much higher computational burden. The detailed modeling methods comparison can be found Appendix C.

The following two theorems show the complexity of the scheduling problem and its optimality condition, which will be used to develop the LW heuristic in section 2.5.

Theorem 2.1. *The 3D AS/RS retrieval task scheduling problem is NP-hard.*

See Appendix D for its proof.

Theorem 2.2: Optimality Condition. *Consider the m^{th} and $(m+1)^{th}$ tasks in an optimal task schedule. If the m^{th} task is for retrieving the $(q_1)^{th}$ SKU from lane j, the $(m+1)^{th}$ task is for retrieving the $(q_2)^{th}$ SKU from lane k, and the $(m-1)^{th}$ task is associated with lane i, we should always have $p_m \geq s_m$ where $p_m = max\{r_{k,q_2} - t_{m-1} - p^3_{ik}, 0\}$ and $s_m = max\{r_{j,q_1} - t_{m-1} - p^3_{ij}, 0\}$ if the $(m-1)^{th}$ task is a retrieval task, otherwise; $p_m = max\{r_{k,q_2} - t_{m-1} - p^2_{i,l} - p^2_{l,k}, 0\}$ and $s_m = max\{r_{j,q_1} - t_{m-1} - p^2_{i,l} - p^2_{l,j}, 0\}$ if the $(m-1)^{th}$ task is to reallocate a shuttle from lane i to l.*

See Appendix D for its proof.

2.5 Solution Methods for 3D AS/RS Retrieval Task Scheduling

The proposed MIP was solved with the Gurobi solver on a Dell desktop with an Intel Core i5 3.40 GHz processor and 8 GB RAM using Windows 10. The computational time of Gurobi increased exponentially as the problem size became larger. Therefore, the Lowest-Waiting-Time-First (LW) heuristic was developed based on Theorem 2.2. Genetic Algorithm (GA) and other two heuristics found from the literature were also adapted to that problem. It is worth to note that we also tried adding additional constraints sets to the MIP model (2.1-2.21) based on Theorem 2.2, but it did not improve the computational efficiency a lot.

2.5.1 First-Come-First-Serve

First-come-first-serve (FCFS) is the rule most frequently used in practice. When a shuttle is ready to hand over an SKU to the crane or be transferred to another lane, it will send a request to the crane. After receiving the request, the crane will travel to that shuttle if it is idle. Otherwise, the request will be added into a waiting list, and the crane will take care of the requests in the waiting list following the FCFS rule. To improve the performance of FCFS, in addition to receiving requests, we allow the crane, after finishing one task, to be able to calculate ready times for all tasks and serve the earliest if the waiting list is empty. When the crane arrives, if the shuttle is ready, the crane will pick up the SKU or shuttle. Otherwise, the crane will wait until the shuttle or SKU is available. If different shuttles/tasks are predicted to be ready at the same time, the crane will choose the target lane randomly. Moreover, for a shuttle reallocation task, the lane that will receive the shuttle is randomly selected among all lanes currently without a shuttle.

The makespan for the crane to finish a given batch of retrieval tasks includes three components: time for transferring SKUs to I/O point (T_r), time for moving shuttles (T_m), and time to wait for an available SKU/shuttle (T_w). Among those three components, T_r is constant, so the scheduling problem aims to minimize T_m and T_w. However, instead of focusing on the crane, the FCFS aims to minimize the waiting time of transactions (i.e., SKUs and shuttles) and can hardly achieve shorter T_m and T_w.

2.5.2 Lowest-Waiting-Time-First

The Lowest-Waiting-Time-First (LW) rule was developed based on Theorem 2.2 aiming to reduce the crane's waiting time T_w. At the beginning of the m^{th} SC, LW calculates the potential

crane's waiting time $PW_{m,i} \geq 0$ for each lane i if the crane is going to serve a task from that lane, and selects the lane with the lowest $PW_{m,i}$ among all lanes. Moreover, whenever a shuttle reallocation task is conducted, the lane with the shortest travel time is selected to receive the shuttle. However, if the target lane is selected only based on the value of $PW_{m,i}$, the crane will tend to finish tasks from lanes far from the I/O point first and take care of the tasks from lanes closer to the I/O point later, which may eventually increase T_w. Therefore, if multiple lanes have the lowest $PW_{m,i}$, the crane will always go to the lane closest to the I/O point. In addition, shuttles needed to be reallocated as early as possible so that tasks from lane $j \in N^1$ can be started earlier. Therefore, a percentage priority ρ is given to shuttle reallocation tasks.

$FT_{m,i}$ is defined as the number of tasks finished on lane i at the beginning of m^{th} SC, where $0 \leq FT_{m,i} \leq Q_i + 1$. N_m^0 and N_m^1 are the set of lanes with a shuttle and the set of lanes need a shuttle at the beginning of the m^{th} SC, respectively. Moreover, $C_m^s = \{i | i \in N_m^0, FT_{m,i} = Q_i\}$ is defined as the set of lanes that have a shuttle ready for reallocation at the beginning of the m^{th} SC. Please note that we only have to track C_m^s when $|N_m^1| > 0$. At last, C_m^p is defined as the set of lanes with a shuttle and retrieval tasks and the lowest $PW_{m,i}$.

The LW can be implemented with the following steps.

Step 1 *(Initialization):* set $m = 1$ and give the initial set of lanes with a shuttle: N_1^0 and the set of lanes requiring a shuttle: N_1^1. For all $i \in N_1^1, PW_{1,i} = \infty$, and calculate $PW_{1,i}$ for $i \in N_1^0$.

Step 2 *(Checking m):* if $m = M + 1$, stop and output T_M, otherwise; go to Step 3.

Step 3 *(Target lane selection):* select the lane with the lowest $PW_{m,i}$ for $i \in N_m^0$. If $|C_m^s| > 0$, with probability ρ, select $i \in C_m^s$ with the lowest $P_{i_0,i}$, go to Step 4, and with probability $1 - \rho$, select $i \in C_m^p$ with the lowest $P_{i_0,i}$, go to Step 5. If $|C_m^s| = 0$, select $i \in C_m^p$ with the lowest $P_{i_0,i}$, go to Step 5.

Step 4 *(Shuttle reallocation):* select lane $j \in N_m^1$ with the lowest $P_{i,j}^2$. Conduct the reallocation task. $FT_{m+1,i} = Q_i + 1$, $N_{m+1}^1 = N_m^1 - \{j\}$. $N_{m+1}^0 = N_m^0 - \{i\} + \{j\}$. Calculate T_m *and* $PW_{m+1,i}$ for $i \in N_{m+1}^0 \cup N_{m+1}^1$, and go to Step 2.

Step 5 *(SKU retrieval):* conduct the retrieval task, $FT_{m+1,i} = FT_{m,i} + 1$, calculate T_m and $PW_{m+1,i}$ for $i \in N_{m+1}^0 \cup N_{m+1}^1$, go to Step 2.

The value of probability ρ depends on the following scenarios, 1) If $|C_m^s| > 0$ and $\min_{i \in C_m^p} PW_{m,i} = 0$, $\rho = |N_m^1|/|N|$; otherwise, 2) if $|C_m^s| > 0$ and $\min_{i \in C_m^p} PW_{m,i} > 0$, $\rho = 1$. For easier tracking $PW_{m,i}$, $L_{m,i}$ can be defined to show the location of SKUs/shuttles on z coordinates (in terms of time) at the beginning of the m^{th} cycle. At Step 1, $L_{1,i} = \begin{cases} r_{i,1} & i \in N_1^0 \\ \infty & i \in N_1^1 \end{cases}$, and $PW_{1,i} = \max\{L_{1,i} - P_{0,i}, 0\}, i \in N_1^0$. If the m^{th} task is for retrieving the q^{th} SKU from lane j, then $L_{m+1,i} = \begin{cases} max\{r_{i,q+1} - p_j^4, 0\} & i = j, \\ \max\{L_{m,i} - P_{0,j} - p_j^4, 0\} & i \in N_{m+1}^0, i \neq j, \text{ and } PW_{m+1,i} = \max\{L_{m+1,i} - P_{0,i}, 0\}. \text{ If the } m^{th} \\ \infty & i \in N_{m+1}^1 \end{cases}$ task is for transferring a shuttle from lane j to lane l, $L_{m+1,i} = \begin{cases} r_{i,1} & i = l, \\ \max\{L_{m,i} - P_{0,j} - p_{j,l}^2, 0\} & i \in N_{m+1}^0, i \neq l, \text{ and } PW_{m+1,i} = \max\{L_{m+1,i} - P_{l,i}^2, 0\}. \\ \infty & i \in N_{m+1}^1 \end{cases}$

2.5.3 Percentage Priority to Shuttle Reallocation with the Shortest Leg (PPS-SL)

Except for the aforementioned two heuristic algorithm, the Percentage Priority to Retrievals with the Shortest Leg (PPR-SL) rule proposed by Yu and De Koster (2012) for scheduling storage and retrieval tasks in 3D AS/RS with conveyor-based DMMs was adapted as Percentage Priority to Shuttle Reallocation with the Shortest Leg (PPS-SL) for this study. PPS-SL, at the beginning of each cycle, selects the lane with the lowest $leg_{m,i}$, where

$$leg_{m,i} = \begin{cases} P_{i_0,i} + PW_{m,i} & i \in |N_m^0|, i \notin |C_m^s|, \\ P_{i_0,i} + \min_{j \in N_m^1} (P_{i,j}^2 + P_{j,1}^1) & i \in |C_m^s|, \\ \infty & i \in N_{m+1}^1. \end{cases}$$

We tried to reallocate shuttles as early as possible. Whenever the $leg_{m,i}\ for\ i \in |C_m^s|$ is updated, with a probability $\rho = |N_m^1|/|N|$, a lower value is assigned (i.e., $leg_{m,i} = \min(P_{i,j}^2 | j \in N_m^1)$) so that shuttle reallocation tasks from $i \in |C_m^s|$ has a higher probability to be conducted earlier.

In PPS-SL, the crane finishes one retrieval task in each cycle, which implies that the crane will wait at lane j for the retrieved SKU in lane j instead take care of a task from another lane after reallocating a shuttle to lane j. The procedure of PPS-SL is very similar to LW and is listed below.

Step 1 *(Initialization):* set $m = 1$ and give the initial set of lanes with a shuttle: N_1^0 and the set of lanes needs a shuttle: N_1^1. For all $i \in N_1^1, leg_{1,i} = \infty$, and calculate $leg_{1,i}$ for $i \in N_1^0$.

Step 2 *(Checking m):* if $m = M - |N^1| + 1$, stop and output T_M, otherwise; go to Step 3.

Step 3 *(Target lane Selection):* select the lane with the lowest $leg_{m,i}$. If $i \in C_m^s$, go to Step 4; Otherwise, go to Step 5.

Step 4 (Shuttle reallocation): Select lane $j \in N_m^1$ with the lowest $P_{i,j}^2 + P_{j,1}^1$. Reallocating shuttle from land i to lane j and retrieve the first SKU from lane j. $FT_{m+1,i} = Q_i + 1, FT_{m+1,j} = 1, N_{m+1}^1 = N_m^1 - \{j\}, N_{m+1}^0 = N_m^0 - \{i\} + \{j\}$. Calculate $T_m, leg_{m+1,i}$ for $i \in N_{m+1}^0 \cup N_{m+1}^1$, and go to Step 2.

Step 5 (SKU retrieval): Conduct the retrieval task, $FT_{m+1,i} = FT_{m,i} + 1,$ calculate $T_m,$ and $leg_{m,i}$ for $i \in N_{m+1}^0 \cup N_{m+1}^1$, and go to Step 2.

2.5.4 Genetic Algorithm

The three aforementioned algorithms are simple heuristics based on specific crane and shuttle dispatching rule and may end at a local optimum with a very high probability. Therefore, the Genetic Algorithm (GA) that begins with a set of initial feasible solutions and searches the feasible region following natural selection and evolution strategies was adopted to explore bigger solution space to escape local optimum.

GA has been commonly used to solve the task scheduling problem of 2D AS/RS (Asokan, P., Jerald, J., Arunachalam, S., & Page, 2008; Noorul Haq et al., 2003; Popović et al., 2014). However, the application of the GA to a 3D system is more difficult for the following reasons. In a 2D AS/RS, chromosomes only represent the sequence of tasks and each chromosome is feasible. In the 3D AS/RS retrieval task scheduling problem, only a "reasonable" shuttle assignment and crane route combination forms a chromosome. Here, a reasonable assignment means that a shuttle should not be assigned to a lane that has already been assigned with another shuttle and should not be moved out of a lane until all retrieval tasks in that lane are finished. A reasonable route means that the crane will not visit a lane if there is no shuttle in that lane unless the crane is moving a shuttle to that lane. Therefore, it is impossible to create a chromosome with an arbitrary schedule of retrieval and shuttle reallocation tasks. In our study, a new chromosome structure is proposed to make the GA applicable to our scheduling problem and a repair strategy is applied to guarantee the feasibility of chromosomes after each operation.

To generate the initial feasible solutions, short-form chromosomes are created first to represent the shuttle assignment. All shuttles are indexed with $i \in \{1, \dots, |N^0|\}$ based on their initial location/lane. In a short-form chromosome, each gene corresponds to a lane. The gene of each lane in N^0 (i.e., each lane with a shuttle at the beginning) is the index of the shuttle at that lane. The gene of each lane in $|N^1|$ is comprised by two parts, the integer and fractional parts. The integer part, ranging from 1 to $|N^0|$, represents which shuttle is assigned to that lane. The fractional part is originally randomly generated in range $(0, 1)$ without any repeats and represents the schedule of shuttle movements. Lanes with the same integer part i will get served by shuttle i based on an ascending order of their fractional parts.

In a chromosome (long-form), each gene represents a lane and will be assigned with a list of numbers. For each number assigned to a lane, the integer part ranging from 1 to $|N^0|$ indicates the shuttle assigned to this lane. A fractional part of each number is generated randomly in the range of 1×10^{-D} to $M \times 10^{-D}$, where M is the number of total SC tasks and D is $\lceil \log_{10} M \rceil$. A lower fractional part means that the corresponding task will be executed by the crane earlier. An example of the short-form and long-form chromosomes is showed in Appendix E. It is also worth to note that we do not have to record the short-from chromosome because we can always extract the short-from chromosome from a long-form chromosome for the following operations. The tournament selection is applied to select parents for the following GA procedures, and multi-point-based crossover is applied for selected chromosomes (Vose, 1999). The mutation procedure is applied on the corresponding short-form chromosome to change the shuttle-to-lane assignment. A repair strategy is applied to guarantee the feasibility of all offspring. The details and examples of these operations can be found in Appendix E.

2.6 Numerical Experiments and Discussion

2.6.1 Comparison of Solution Methods

Numerical experiments were conducted to validate the MIP model and test the efficiency of the four proposed solution methods. Gurobi was used to solve the MIP model for small cases with a running time limit of 600 seconds. When it reaches the time limitation, Gurobi will stop and return a feasible solution if it can find one. Moreover, the population size, crossover rate, and mutation rate for GA are set as 10, 0.8, and 0.08 respectively. The GA will stop if there is no improvement in 100 generations, or the running time exceeds one hour. The 3D AS/RS

considered for this analysis is a real-world rack. The rack has three layers, 80 columns, and each lane has 15 storage cells. The total number of storage lanes is 240, and the rack can store 3,600 SKUs. The length (*x*-direction) and width (*z*-direction) of a storage cell are all 1.4 meters, and the height (y direction) of a storage cell is 2 meters. The maximum velocity of shuttles (V_s) is $1.5m/s$,and the acceleration/deceleration (a_s) is $1m/s$. The maximum velocity for the crane in *x*-direction and *z*-direction are $V_c^x = 2.5m/s \; and \; V_c^z = 0.5m/s$, respectively. The acceleration/deceleration in the *x*-direction and *z*-direction are $a_c^x = 0.5m/s^2$ and $a_c^z = 0.5m/s^2$, respectively. Time consumed for the crane or shuttles to load/unload an SKU is $1s$.

Table 2-1 shows 10 small cases with different values of $|N^0|$ and $|N|$, and Q_i for the comparison among solving MIP by Gurobi and the four proposed solution methods. Four deviations, namely $\Delta_{FCFS} = \frac{T_{FCFS} - T_{MIP}}{T_{MIP}}$, $\Delta_{LW} = \frac{T_{LW} - T_{MIP}}{T_{MIP}}$, $\Delta_{PPS-SL} = \frac{T_{PPS-SL} - T_{MIP}}{T_{MIP}}$ and $\Delta_{GA} = \frac{T_{GA} - T_{MIP}}{T_{MIP}}$, are used to show the difference, where T_{MIP}, T_{FCFS}, T_{LW}, T_{PPS-SL}, and T_{GA} are the makespan of solutions generated by solving MIP with Gurobi, FCFS, LW, PPS-SL, and GA, respectively. $Time_{MIP}$ and $Time_{GA}$ denote the running time of Gurobi and GA in seconds. Because the FCFS, LW, and PPS-SL are simple heuristics, the running time is much shorter than one second and considered ignorable. Moreover, N/A represents that Gurobi could not find a feasible solution within the time limitation.

As Table 2-1 shows, Gurobi can solve MIP (2.1-2.21) to optimum when the problem size is small. However, the solving time increase dramatically in $\sum_{i \in N} Q_i$, $|N^0|$, and $|N|$, and a feasible solution cannot even be discovered as the problem size getting big (e.g., (103,17,4)). GA can usually yield a solution with a shorter makespan than other heuristics (i.e., FCFS, LW, and PPS-SL). Even though GA cannot guarantee the optimality, it has a higher probability to reach the global optimum. LW is better than FCFS and PPS-SL for solution quality. PPS-SL is the worst among the four proposed algorithms since it makes the crane wait for the SKU in a lane after reallocating a shuttle to that lane. The numerical experiments were also conducted for large cases to further demonstrate the efficiency of the proposed algorithms.

Six large cases were considered with the number of retrieval tasks ($\sum_{i \in N} Q_i$) ranging from 600 to 3,600 with the step size of 600. Each case consists of 70 sub-cases in terms of 10 different number of shuttles ($R = |N^0|/|N| \; = 0.1, 0.2, \cdots, 1$) and seven shuttle speed profiles. Table 2-2

Table 2-1. Comparison among MIP, FCFS, LW, PPS-SL, and GA for Smaller Cases

$(\sum_{i\in N} Q_i, \lvert N \rvert, \lvert N^0 \rvert)$	$Time_{GA}$	$Time_{MIP}$	Δ_{FCFS}	Δ_{LW}	Δ_{PPS-SL}	Δ_{GA}
$(7, 6, 5)$	10	3	3.9%	0.6%	39.4%	0%
$(83, 11, 9)$	40	600	-2.4%	-2.5%	-1.4%	-2.9%
(103,17,4)	179	600	N/A(0%)*	N/A(-2.7%)	N/A(-0.3%)	N/A(-2.3%)
$(53, 19, 14)$	92	600	9.6%	0.1%	9.3%	-1.4%
$(32, 19, 3)$	228	600	-10.6%	-19.9%	0.2%	-2.9%
$(38, 13, 8)$	66	600	10.3%	-2.4%	18.0%	-1.4%
$(36, 8, 7)$	16	600	11.2%	6.6%	8.6%	-2.9%
$(15, 10, 3)$	21	600	-3.1%	-5.2%	46.1%	-1.4%
(14,11, 8)	26	600	20.4%	7.1%	43.2%	-2.9%
$(12, 8, 6)$	15	600	12.0%	9.9%	36.7%	1.1%

* N/A means that Gurobi failed to find a feasible solution and the listed percentages are the differences from FCFS.

Table 2-2. Four Heuristics (LW, PPS-SL, GA, and GA_{LW}) against FCFS for Large Cases

$V_s\ (m/s)$		0.07				0.44				3.04			
$a_s(m/s^2)$		0.06				0.24				1.06			
$\sum_{i\in N} Q_i$	R	Δ'_{LW}	Δ'_{PPS-Sl}	Δ'_{GA}	$\Delta'_{GA_{LW}}$	Δ'_{LW}	Δ'_{PPS-Sl}	Δ'_{GA}	$\Delta'_{GA_{LW}}$	Δ'_{LW}	Δ'_{PPS-Sl}	Δ'_{GA}	$\Delta'_{GA_{LW}}$
600	0.1	-17.4%	90.9%	0.0%	-17.4%	-17.4%	11.6%	-1.6%	-17.5%	-19.0%	-13.6%	-3.5%	-19.0%
	0.2	-15.7%	85.8%	-0.1%	-15.7%	-15.4%	9.0%	-2.2%	-15.4%	-15.1%	-10.4%	-2.9%	-16.1%
	0.3	-16.5%	69.6%	0.0%	-16.5%	-14.0%	7.8%	-3.3%	-14.0%	-14.0%	-11.2%	-3.4%	-14.6%
	0.4	-11.8%	70.5%	-0.1%	-11.9%	-12.3%	5.1%	-4.2%	-12.3%	-10.9%	-8.7%	-4.0%	-11.2%
	0.5	-10.3%	58.8%	-0.1%	-10.4%	-10.0%	5.0%	-4.1%	-10.4%	-10.4%	-8.7%	-4.8%	-10.6%
	0.6	-8.0%	41.4%	-3.5%	-8.1%	-8.7%	5.3%	-3.6%	-8.8%	-8.5%	-6.7%	-4.3%	-8.8%
	0.7	-7.8%	31.7%	-3.3%	-8.0%	-6.2%	3.1%	-4.6%	-6.4%	-7.0%	-5.9%	-4.8%	-7.0%
	0.8	-4.5%	20.4%	-3.3%	-4.7%	-4.0%	3.7%	-2.5%	-4.2%	-5.0%	-4.0%	-4.2%	-5.2%
	0.9	-2.9%	13.6%	-3.1%	-2.9%	-2.8%	0.8%	-2.9%	-2.9%	-2.5%	-1.9%	-2.3%	-2.5%
	1	-0.9%	-0.8%	0.0%	-0.9%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
1,800	0.1	-7.4%	23.6%	0.0%	-7.4%	-7.0%	2.7%	0.0%	-7.0%	-7.1%	-5.4%	-1.4%	-7.1%
	0.2	-5.7%	20.5%	0.0%	-6.1%	-6.0%	0.7%	0.0%	-6.1%	-6.3%	-5.3%	-1.2%	-6.3%
	0.3	-5.4%	17.5%	0.0%	-5.4%	-5.2%	0.2%	-0.6%	-5.3%	-5.5%	-4.8%	-1.7%	-5.5%
	0.4	-3.4%	13.9%	0.0%	-3.5%	-4.3%	0.7%	0.0%	-4.3%	-4.5%	-4.0%	-1.9%	-4.5%
	0.5	-3.4%	12.2%	0.0%	-3.6%	-3.7%	0.6%	-1.3%	-3.7%	-4.1%	-3.6%	-1.8%	-4.1%
	0.6	-3.0%	9.3%	0.0%	-3.1%	-2.7%	0.5%	-1.3%	-2.7%	-2.7%	-2.5%	-2.0%	-2.7%
	0.7	-2.3%	5.8%	0.0%	-2.3%	-2.3%	0.3%	-1.1%	-2.3%	-2.3%	-2.1%	-1.5%	-2.3%
	0.8	-1.3%	4.2%	-0.4%	-1.5%	-1.9%	-0.5%	-1.4%	-1.9%	-1.4%	-1.3%	-1.5%	-1.4%
	0.9	-0.9%	2.1%	-0.9%	-1.3%	-0.7%	0.2%	-0.7%	-0.7%	-0.9%	-0.8%	-0.9%	-0.9%
	1	-0.1%	0.0%	0.0%	-0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
3,000	0.1	-3.5%	8.6%	0.0%	-3.5%	-4.1%	-0.5%	0.0%	-4.2%	-4.6%	-4.3%	-1.0%	-4.6%
	0.2	-2.7%	7.4%	0.0%	-2.9%	-3.3%	-0.7%	0.0%	-3.3%	-3.9%	-3.9%	-0.5%	-3.9%
	0.3	-2.9%	4.5%	0.0%	-3.0%	-3.1%	-1.0%	-0.1%	-3.1%	-3.4%	-3.4%	-1.1%	-3.4%
	0.4	-1.8%	4.2%	0.0%	-2.0%	-2.9%	-1.3%	-0.2%	-2.9%	-3.1%	-3.1%	-1.0%	-3.1%
	0.5	-1.6%	3.2%	0.0%	-1.8%	-2.2%	-0.9%	-0.2%	-2.2%	-2.2%	-2.2%	-1.2%	-2.2%
	0.6	-1.3%	2.5%	0.0%	-1.4%	-1.6%	-0.6%	-0.8%	-1.6%	-2.1%	-2.1%	-0.9%	-2.1%
	0.7	-0.8%	2.5%	-0.2%	-0.9%	-1.3%	-0.6%	-0.6%	-1.3%	-1.6%	-1.5%	-1.1%	-1.5%
	0.8	-0.3%	1.2%	0.0%	-0.5%	-0.9%	-0.4%	-0.7%	-0.9%	-0.9%	-0.9%	-0.7%	-0.9%
	0.9	-0.2%	1.2%	-0.4%	-0.2%	-0.4%	-0.2%	-0.4%	-0.4%	-0.5%	-0.5%	-0.5%	-0.5%
	1	0.0%	1.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

shows the results of cases with 600, 1,800 and 3,000 retrieval tasks under three speed profiles. The impact of starting GA with an LW solution is also tested. Four deviations from FCFS, namely $\Delta'_{LW} = \frac{T_{LW} - T_{FCFS}}{T_{FCFS}}$, $\Delta'_{PPS-SL} = \frac{T_{PPS-SL} - T_{FCFS}}{T_{FCFS}}$, $\Delta'_{GA} = \frac{T_{GA} - T_{FCFS}}{T_{FCFS}}$ and $\Delta'_{GA_{LW}} = \frac{T_{GA_{LW}} - T_{FCFS}}{T_{FCFS}}$, are used to show the difference, where $T_{GA_{LW}}$ is the shortest makespan found by GA staring with the LW solution in the first generation. Table 2-2 shows that LW outperforms the other algorithms except GA_{LW} and yields solutions better than FCFS by up to 19%. Under a given speed profile and $\sum_{i \in N} Q_i$, the advantage of LW against FCFS reduces as more shuttles are added into the system. As discussed in subsection 4.1, the crane's waiting time T_w and distance for reallocating shuttles T_m are expected to be reduced through a better scheduling. However, when the system has more shuttles, which can work simultaneously in the system, LW, compared against FCFS, cannot reduce T_w by much as there is a higher probability to make the crane busy all the time. Meanwhile, having more shuttles means fewer shuttle reallocations so that LW also cannot reduce T_m, compared against FCFS by much. Table 2-2 also shows that, for given R, V_s and a_s, $|\Delta'_{LW}|$ reduces as $\sum_{i \in N} Q_i$ gets larger. The reason is that having more tasks makes T_r more dominant among the three time components for the crane. Even though having a larger $\sum_{i \in N} Q_i$ will not hurt the advantage of LW in terms of T_w and T_m, $|\Delta_{LW}|$ in percentage decreases because of the much higher total makespan.

Moreover, when shuttles are slow, PPS-SL is the worst among all proposed algorithms. Even though PPS-SL is designed to reduce the crane's waiting time when it takes care of the retrieval tasks, it sacrifices the opportunity for the crane to take care of tasks from other lanes after reallocating a shuttle. When shuttles are slow, PPS-SL results in much higher T_w than FCFS. However, as shuttles become faster, the drawback of PPS-SL decreases and can eventually outperforms FCFS when shuttles are fast enough (e.g., $V_s = 0.44\ m/s$ and $a_s = 0.24\ m/s^2$). Furthermore, having more shuttles means fewer shuttle reallocations and can reduce the disadvantage of PPS-SL. Different from small cases, GA can only outperform FCFS by at most 4.8% and is much worse than LW for large cases. Even when we started GA with the LW solution, it can only provide a solution slightly better than LW. It means either that LW has already yielded a solution with high quality or that GA is likely to be trapped at a local optimal solution (e.g., the LW solution).

It seems to contradict the statement we made in section 2.3 that 'GA is commonly used and can usually provide a good solution in a reasonable amount of time'. As demonstrated by Table 2-1. Comparison among MIP, FCFS, LW, PPS-SL, and GA for Smaller Cases, GA can yield better solutions than other heuristics for most of the small cases. When large cases are tested, however, GA has to explore a significantly large solution space and check the feasibility of offsprings and correct them twice (i.e., after crossover and mutation) in each iteration, which takes much time. However, we limit the total running time to one hour, so GA cannot explore a large enough solution space to well beat LW, which is developed based on the optimality condition. If we relax the limitation on the total running time and population size, which is not applicable in practice, we expect that GA can yield better solutions. Moreover, the statement that 'GA is commonly used and can usually provide a good solution in a reasonable amount of time' was made based on the literature review of tasks scheduling problems for 2D AS/RS. When GA is applied to a 2D system, there is no need to check the feasibility of offsprings so that GA can search a large feasible region very fast and potentially beat other heuristics. However, when we apply GA to a 3D system, the computational burden is significantly increased.

2.6.2 Improvement of 3D AS/RS Performance

Based on the comparison among the proposed solution methods, we consider LW the best based on its solving time and solution quality. Therefore, more numerical experiments based on LW were conducted to explore the desired number of shuttles in a system. Considering the real-world rack in our study, we examined the system performance under different numbers of shuttles (R) and three equipment speed profiles S_i, where $i = 1, 2, \text{and } 3$. Three speed profiles are $S_1 = (V_c^x = 2.5m/s\,, V_c^z = 0.5m/s, a_c^x = 0.5m/s^2, a_c^z = 0.5m/s^2\,, V_s = 1.5m/s, a_s{=}1m/s^2\,)$, $S_2 = (V_c^x = 2.5m/s\,, V_c^z = 0.5m/s, a_c^x = 0.5m/s^2, a_c^z = 0.5m/s^2\,, V_s = 3m/s, a_s{=}2m/s^2\,)$ and $S_3 = (V_c^x = 5m/s\,, V_c^z = 1m/s, a_c^x = 1m/s^2, a_c^z = 1m/s^2\,, V_s = 1.5m/s, a_s{=}1m/s^2\,)$. Here, the speed profile S_1 is the base case while S_2 and S_3 are for faster shuttles and faster crane, respectively. Because not all lanes have retrieval tasks in a planning horizon, we considered only 100 lanes in Table 2-3, which shows the system performance (T_{LW}) in hours under different combinations of R, S_i and $\sum_{i \in N} Q_i$. For each combination, three replications in terms of different locations of tasks and shuttles were considered, and Table 2-3 shows the average of the three results.

Table 2-3. Makespan $T_{LW}(\sum_{i\in N} Q_i, R)$ in Hours for Instances with 100 Lanes and up to 1,000 Tasks

R	$\sum_{i\in N} Q_i$	100	200	300	400	500	600	700	800	900	1,000
0.1	S_1	1.47	2.77	4.09	5.40	6.75	8.04	9.31	10.62	11.93	13.25
	S_2	1.45	2.76	4.09	5.40	6.75	8.04	9.30	10.63	11.94	13.24
	S_3	0.84	1.56	2.31	3.03	3.77	4.49	5.20	5.94	6.67	7.40
0.2	S_1	1.43	2.75	4.09	5.40	6.74	8.04	9.30	10.62	11.93	13.23
	S_2	1.42	2.75	4.09	5.40	6.74	8.04	9.29	10.62	11.93	13.23
	S_3	0.80	1.55	2.29	3.01	3.76	4.49	5.20	5.93	6.66	7.39
0.3	S_1	1.40	2.75	4.08	5.40	6.73	8.02	9.29	10.61	11.92	13.22
	S_2	1.39	2.75	4.08	5.40	6.73	8.02	9.29	10.61	11.91	13.22
	S_3	0.79	1.54	2.28	3.02	3.76	4.49	5.20	5.92	6.66	7.38
0.4	S_1	1.38	2.73	4.08	5.39	6.72	8.01	9.28	10.59	11.90	13.21
	S_2	1.38	2.74	4.08	5.39	6.72	8.01	9.28	10.59	11.90	13.21
	S_3	0.78	1.52	2.28	3.02	3.76	4.48	5.19	5.92	6.65	7.38
0.5	S_1	1.37	2.72	4.06	5.37	6.71	8.00	9.27	10.58	11.89	13.20
	S_2	1.37	2.72	4.06	5.37	6.70	7.99	9.27	10.58	11.89	13.20
	S_3	0.77	1.52	2.26	3.00	3.74	4.47	5.18	5.91	6.64	7.38
0.6	S_1	1.36	2.70	4.04	5.36	6.69	7.98	9.25	10.57	11.87	13.18
	S_2	1.35	2.70	4.04	5.36	6.69	7.98	9.26	10.57	11.87	13.18
	S_3	0.76	1.51	2.25	2.99	3.73	4.46	5.16	5.90	6.63	7.36
0.7	S_1	1.34	2.68	4.02	5.34	6.67	7.96	9.23	10.55	11.86	13.17
	S_2	1.34	2.68	4.02	5.34	6.67	7.96	9.23	10.55	11.86	13.17
	S_3	0.75	1.50	2.25	2.98	3.72	4.44	5.15	5.88	6.62	7.35
0.8	S_1	1.34	2.67	4.00	5.31	6.65	7.94	9.22	10.53	11.84	13.15
	S_2	1.34	2.67	4.00	5.31	6.64	7.94	9.22	10.53	11.84	13.15
	S_3	0.75	1.50	2.24	2.97	3.71	4.44	5.15	5.88	6.61	7.35
0.9	S_1	1.33	2.66	3.99	5.30	6.63	7.93	9.20	10.51	11.82	13.13
	S_2	1.33	2.66	3.99	5.30	6.63	7.93	9.20	10.51	11.82	13.13
	S_3	0.74	1.49	2.23	2.96	3.70	4.43	5.14	5.87	6.61	7.34
1.0	S_1	1.32	2.66	3.98	5.30	6.63	7.92	9.19	10.51	11.82	13.13
	S_2	1.32	2.66	3.98	5.30	6.63	7.92	9.19	10.51	11.82	13.13
	S_3	0.74	1.49	2.23	2.96	3.70	4.43	5.13	5.87	6.60	7.33

Table 2-3 shows that T_{LW}, for any S_i, increases almost linearly as the value of $\sum_{i\in N} Q_i$ increases. Let us define $T_{LW}(\sum_{i\in N} Q_i, R)$ as the makepan for finishing $\sum_{i\in N} Q_i$ retrieval tasks. $T_{LW}(\sum_{i\in N} Q_i, 1)$ is approximately equals to $\frac{\sum_{i\in N} Q_i}{100} \times T_{LW}(100, 1)$. For $R < 1$, $T_{LW}(\sum_{i\in N} Q_i, R)$ is approximately equals to $T_{LW}(100, R) + \frac{\sum_{i\in N} Q_i - 100}{100} \times b$, where $b \approx T_{LW}(100, 1)$. This approximate relationship implies that when the number of shuttles in the system is fixed, time for moving shuttles, T_m, is almost constant and roughly equal to $T_{LW}(100, 1) - T_{LW}(100, R)$. In addition, for given speed profile S_i, the marginal benefit in terms of makespan of adding shuttles decreases as the value of R gets larger. The marginal benefits of having more shuttles are also independent from $\sum_{i\in N} Q_i$. For instance, under speed profile S_1, adding 90 more shuttles into the system with 10 shuttles (i.e., increasing R from 0.1 to 1) can at most reduce makespan by $0.14\ hour$ no matter how many retrieval tasks are scheduled in a planning horizon. However, as $\sum_{i\in N} Q_i$ increased, T_{LW} increases significantly. In other words, having more tasks means a greater makespan but does not necessarily imply the need for more shuttles. Table 2-3 also shows the impact on system performance by changing the equipment's speed. S_2 doubles shuttle maximum velocity and acceleration/deceleration in the base speed profile S_1. The makespan under these two speed profiles are quite similar, especially when there are many shuttles. However, the speed profile S_3, which doubles the maximum velocity and acceleration/deceleration of cranes, significantly reduces the makespan compared to the speed profile S_1. As discussed above, for cases with a large number of tasks, time for moving SKUs, T_r, dominates the other two components that form the makespan. Therefore, increasing the crane's speed and acceleration can reduce both T_r, and time for moving shuttles, T_m, while higher shuttle speed can only affect T_w. Especially when shuttles are already fast enough (S_1) or R is great enough to keep the crane busy, having faster shuttles may not help makespan reduction much.

In 3D AS/RS practice, lead time (i.e., responsiveness) and makespan (i.e., efficiency) are two important metrics for evaluating operational performance. A company can reduce its planning horizon to be more responsive (i.e., shorter lead times for orders). A company that promises second day deliveries may have a planning horizon of 12 or 24 hours. If the company wants to have same-day delivery, it may reduce the planning horizon to several hours and solve the scheduling problem discussed in this study more frequently, with smaller batches. However, being responsive (i.e., more frequent scheduling) may sacrifice the opportunity to reduce the

makespan. Therefore, different strategies should be adopted under different business scenarios. As mentioned before, adding more shuttles makes a big improvement of system performance only when there are fewer retrieval tasks in one planning horizon. For a warehouse system with a higher responsiveness requirement, such as a local fulfillment center for last-mile delivery, a short planning horizon implies fewer orders in one scheduling problem (e.g., 200 retrieval tasks in 2~3 hours). More shuttles can be added into the system to reduce makespans. On the other hand, a warehouse with a lower responsiveness requirement may reduce planning frequency and schedule more tasks in a longer planning horizon. To reduce the makespan or incorporate more retrieval tasks for this case, the warehouse may increase the crane speed and acceleration rather than add more shuttles. Another possibility of reducing the number of lanes served by a crane will have a similar effect to higher crane speed. A cost comparison can be made to facilitate the choice between adding more cranes and adjusting the crane speed.

2.7 Conclusion and Future Research

This study examines the task scheduling problem in a crane-based 3D AS/RS with shuttle-based DMMs. The system allows fewer shuttles than lanes to decrease the cost and improve the system flexibility. For this study, we considered only retrieval tasks executed in the SC mode. Reallocating shuttles and the coordination between different and independent movement elements (i.e., one crane and multiple shuttles) significantly increases modeling and computational efforts compared to other AS.RS (e.g., 2D AS/RS and 3D AS/RS with conveyor-based DMMs). An MIP model was proposed to minimize the makespan for the crane to finish a given set of retrieval tasks. The problem was proven to be NP-hard. Considering the features of the problem, FCFS, PPS-SL, and GA from the literature were adapted to address the computational burden. Moreover, LW was developed according to the optimality condition of the scheduling problem. Small cases were randomly generated to compare the performance of MIP and the four heuristics. The numerical results indicate that the time for Gurobi to solve the MIP increases exponentially in the number of lanes, tasks, and shuttles. PPS-SL, which is adapted from the best heuristic in literature for 3D system with conveyor-based DMMs (Yu and De Koster, 2012), might be the worst when the number of shuttles in the system is small. GA is always better than the other algorithms and has a higher probability of reaching a quality solution for small case problems. However, when the problem size is large enough, like in the practical situation, LW becomes the best considering solution quality and solving time.

Through numerical experiments, we also obtained the following two insights. 1) When facing a higher level of responsiveness, a warehouse system should operate in a short planning horizon with small batch size and improve the system performance by adding more shuttles into the system. 2) When the priority is to reduce the makespan (i.e., improve efficiency), a warehouse system should operate in a longer planning horizon with large batch size and increase the crane speed.

In our study, we considered only retrieval task scheduling with the SC mode. In some industries, storage tasks are also vital. Therefore, in future studies, we will also consider storage tasks and the scheduling problem in the DC mode. In addition, according to our numerical experiments, changing the shuttle speed will not affect system performance very much. However, we predict that the number of shuttles may also have a significant effect on system performance if shuttle speed is low. When there are many SKUs or energy is a big concern, a warehouse may have slow shuttles. The relationship between crane and shuttle speeds should be further studied to provide more insights for 3D AS/RS designs. Furthermore, how the physical design and rack dimensions affect system performance is also an interesting future research question. All of these future research directions will involve great computational challenges. Therefore, innovative heuristics or algorithms may be necessary to address the computational burden.

Besides studying these problems independently, a study of how all design and operational factors work together to affect system performance would be interesting. Since even one specific optimization problem is computationally challenging, a study that considers all factors will be extremely complex. Therefore, based on various optimization models and algorithms that are developed for each decision problem, statistical analyses and sensitivity analyses could be adopted to numerically build functions representing the relationship between design factors and performance metrics (i.e., mechanisms by which different design parameters affect system performance under given environments). Those functions can be used by practitioners to quickly determine if a crane-based 3D AS/RS with shuttle-based DMMs is applicable and, if so, how to

CHAPTER THREE
TRAVEL TIME MODELS FOR TIER-TO-TIER SBS/RS WITH DIFFERENT STORAGE POLICIES AND SHUTTLE DISPATCHING RULES

3.1 Abstract

The performance of a tier-to-tier shuttle-based storage and retrieval system is affected by both storage policy and shuttle dispatching rule. This paper was the first to explore both random and class-based storage policies and three shuttle dispatching rules: 1) random, 2) distance-based, and 3) demand-rate based. Modeling the system as a discrete-time Markov Chain, this study derived the shuttle distribution under each policy combination, and further developed the expected travel time models for both the single-command (SC) and dual-command (DC) operations. The models were validated by simulation. Numerical experiments showed significant impacts of the policy combination on the expected travel time. Class-based storage, if implementable, is always better than random storage no matter which dispatching rule is adopted. However, there are possibly other storage policies that could be better than both random and class-based storage. Under random storage and an SC cycle, the distance-based shuttle dispatching rule is always better than the random rule, and the benefit increases and then decreases in the number of shuttles. In a DC cycle, depending on equipment speeds, the distance-based shuttle dispatching rule can be worse than the random rule. When classed-based storage is adopted and the number of shuttles is small, the demand-rate-based rule is the best under an SC cycle, but might be worse than the other rules under a DC cycle. Its benefit becomes more obvious when the demand rates become more heterogeneous until some point but decreases as the system has more shuttles. The demand-rate-based rule can be even dominated by the other shuttle dispatching rules when the demand rates are rather homogeneous.

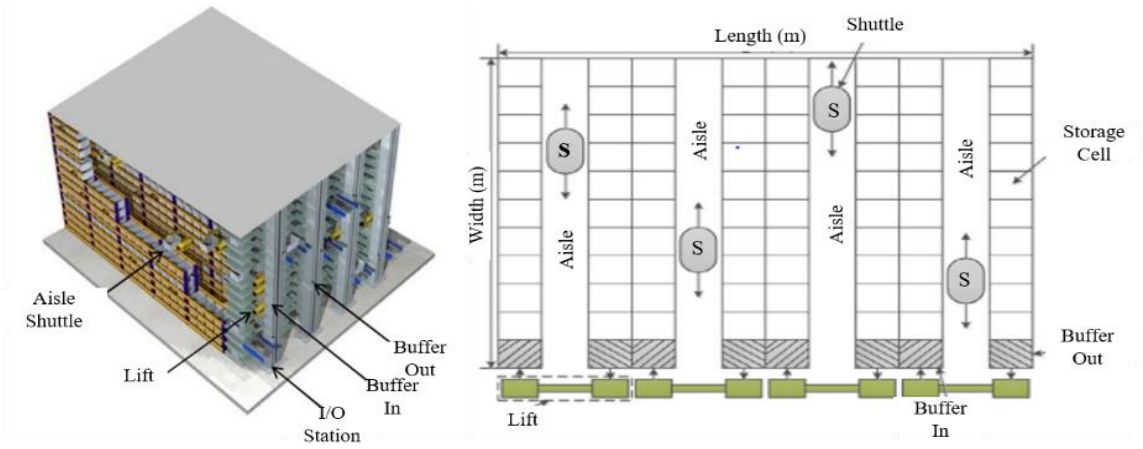
3.2 Introduction

Traditional picker-to-part warehousing systems have been widely replaced by automated material handling systems, such as automated storage and retrieval system (AS/RS), to save labor and space (Azadeh et al., 2019a; Boysen et al., 2019; Liu et al., 2020; Roodbergen and Vis, 2009; Yang et al., 2015b). The main components of AS/RS are storage racks, aisles, storage/retrieval (S/R) machines, and input/output (I/O) stations. Racks are often metal structures with storage cells to accommodate stock-keeping units (SKUs). An S/R machine can travel along an aisle between two racks to pick up or drop off SKUs. An I/O station is where retrieved SKUs are dropped off and incoming SKUs are picked up by an S/R machine. In a traditional AS/RS, aisle-captive cranes, functioning as S/R machines, travel along vertical and horizontal directions

simultaneously. However, the application of aisle-captive cranes requires a high investment and may reduce system flexibility (Azadeh et al., 2019a, 2019b; Ha and Chae, 2019; Lerher et al., 2021; Tappia et al., 2017).

Instead of cranes, a shuttle-based storage and retrieval system (SBS/RS) uses lifts and shuttles as S/R machines to improve flexibility and productivity. In an SBS/RS, illustrated by Figure 3-1, each aisle is equipped with a lift for vertical movements (Ekren, 2017, 2020; Küçükyaşar et al., 2021). The rack system is divided into tiers, and aisle-captive shuttles (Lerher et al., 2021) are used to move SKUs on each tier. Except for the I/O stations on the ground level, an SBS/RS also consists of buffer positions and roller conveyors on each tier, which enable shuttles and lifts to work independently. For instance, when conducting a retrieval task, a shuttle picks up an SKU, moves it to the buffer position, and then is released for another task. A lift picks up the SKU from the buffer position and travels back to the ground level. Based on the type of shuttles, SBS/RS can be categorized into tier-to-tier SBS/RS and tier-captive SBS/RS. In a tier-to-tier SBS/RS, there are fewer shuttles than tiers so lifts move not only SKUs but also empty shuttles across tiers. In a tier-captive SBS/RS, shuttles are dedicated to tiers so that lifts only take care of SKUs. It is widely recognized that a tier-to-tier SBS/RS lowers hardware cost and provides a higher level of flexibility and reliability as system operations will not be halted because of shuttle failure or maintenance (Azadeh et al., 2019a).

Estimating the expected travel/cycle time is a fundamental step for designing and operating a warehouse system as the expected travel time determines the responsiveness and system throughput capacity (Azadeh et al., 2019b; Roodbergen and Vis, 2009; Yang et al., 2017; Zaerpour et al., 2013, 2017; Zammori et al., 2021). The expected travel time denotes the average amount of time for a transaction to take. Usually, lifts and shuttles in an SBS/RS can operate in two modes: 1) a single-command (SC) Cycle: A lift/shuttle executes one retrieval or storage task in one operational cycle, and 2) a dual-command (DC) Cycle: A lift/shuttle executes one retrieval and one storage task in one operational cycle. Given the fact that retrieval tasks are sometimes more critical in operations, especially under the circumstance of online retailing, which requires high responsiveness (Boysen et al., 2019; Yu and De Koster, 2009b; Zou et al., 2016), and the retrieval and storage tasks can be performed in different time windows (Dong et al., 2021; Liu et al., 2020; Wang et al., 2015), this study first considers the expected travel time for retrieval tasks executed in SC operations. However, in many other industries, retrieval tasks and storage tasks



a. Overview of SBS/RS (Marchet et al. 2012) b. Top view of SBS/RS (Zou et al. 2016)

Figure 3-1. An Overview and Top View of SBS/RS

(e.g., re-storage) are equally important and are conducted simultaneously (Azadeh et al., 2019a; Eder, 2019; Ekren et al., 2018; Schenone et al., 2020). Therefore, this study then considers both retrieval and re-storage tasks in DC operations.

The expected travel time is highly influenced by storage policies (Azadeh et al., 2019a, 2019b; Eder, 2020; Silva et al., 2020). There are two major policies assigning SKUs to storage cells in practice and literature, random storage policy and demand-rate-based storage policy. Under random storage, incoming SKUs are randomly assigned to storage cells. Assigning the SKUs with higher demand rates to storage cells closer to I/O stations may reduce the response time. However, it is unrealistic to assume the demand rate of every SKU in a system to be constant over time (Hausman et al., 1976), so this study considers class-based storage, under which all storage locations and SKUs are partitioned into multiple classes according to travel distances to I/O station and demand rates, respectively. The SKUs belonging to a higher demand rate class are then assigned to the storage locations closer to I/O points. SKUs in the same class are assigned randomly to storage cells within the corresponding class. Moreover, in a tier-to-tier SBS/RS, lifts move shuttles across tiers so the system needs to decide which shuttle should be dispatched when a retrieval task is located at a tier without a shuttle. This study considers three shuttle dispatching rules — random shuttle dispatching rule ($\mathcal{R}_1$), distance-based shuttle dispatching rule ($\mathcal{R}_2$), and demand-rate-based shuttle dispatching rule ($\mathcal{R}_3$). The details of these three dispatching rules are listed in sections 3.4 and 3.5. Both storage policies and three shuttle dispatching rules have been technically realized by a major SBS/RS designer and manufacturer, our industrial partner. They were interested in which combination of storage policy and shuttle rule is the most efficient, which motivated this research.

Even though tier-to-tier SBS/RS has been studied under both random and class-based storage policies in the literature, most existing studies assumed random shuttle dispatching. We expected that storage policies and shuttle dispatching rules would both affect the distribution of shuttles and therefore influence the expected travel time. The interactions between these two policies have never been studied for a tier-to-tier SBS/RS, and therefore, there is a lack of methodologies for calculating travel times of a tier-to-tier SBS/RS characterized under different storage policy, shuttle dispatching rules, and operational cycle (i.e., SC and DC). To bridge this research gap, this study developed the expected travel time models for a tier-to-tier SBS/RS implementing different combination of storage and shuttle dispatching policies in either an SC or DC cycle.

Three questions were answered by this study: 1) What the system performance is under different combinations of storage and shuttle dispatching policies; 2) What is the best storage and shuttle dispatching policy for different business needs characterized by demand pattern and operational cycles; 3) How the number of shuttles impacts the system performance.

A tier-to-tier SBS/RS can be modeled as a continuous-time Markov Chain (CTMC). However, since we only focus on the shuttle distribution instead of transition time, this study only modeled the embedded discrete-time Markov Chain (DTMC) and derived steady-state probabilities for distribution of shuttles under different combinations of storage policies and shuttle dispatching rules. Expected travel time models were created and compared for selecting the best combination of storage policy and shuttle dispatching rule under different scenarios in both SC and DC operations. With the proposed expected travel time models, a warehouse designer or manager can easily compare the performance of different storage policies and shuttle dispatching rules and determine the appropriate number of shuttles and speed of equipment in their system.

The remainder of this chapter is organized as follows. Section 3.3 reviews previous studies related to expected travel time estimation for various SBS/RS and discusses the differences of this study from them. Section 3.4 describes the tier-to-tier SBS/RS and presents travel time models for each combination of storage and shuttle dispatching policies in an SC cycle. Section 3.5 extends models for DC operations. In Section 3.6, the proposed travel time models are validated through simulation and then used to compare different storage policies and shuttle dispatching rules with extensive numerical experiments. Finally, Section 3.7 concludes the study and discusses the future work.

3.3 Literature Review

An SBS/RS is considered a special edition of autonomous vehicle storage and retrieval system (AVS/RS) due to their similar physical configurations (Carlo and Vis, 2012; Ekren, 2017; Ha and Chae, 2019). Numerous studies considering the travel time measurement of both tier-captive and tier-to-tier AVS/RSs have been done through simulation (Ha and Chae, 2018; Marchet et al., 2013) and analytical models (Azadeh et al., 2019b; Cai et al., 2014; D'Antonio and Chiabert, 2019; Fukunari and Malmborg, 2008, 2009; Roy et al., 2017). Different system configurations and operational policies have been considered to provide design and management insights, such as the dimension of rack systems (Kuo et al., 2007; Marchet et al., 2013, 2015; Roy et al., 2012;

Tappia et al., 2017), type and number of S/R machines (Roy et al., 2017) and dwell point policies (Roy et al., 2015). However, these studies are different from our work because of the different operational mechanisms between AVS/RS and SBS/RS.

As mentions in section 3.2, lifts and shuttles can work independently in an SBS/RS because of buffer stations and conveyors. However, most AVS/RS studies assume that shuttles and lifts work in a sequential processing policy, which means a shuttle will not request for a lift until the shuttle gets the ordered SKU, and the shuttle cannot be released until the SKU is picked up (Cai et al., 2014; Kuo et al., 2007). Although Zou et al. (2016) considered the parallel processing policy allowing lifts and shuttles to work simultaneously, but they focused on a tier-captive system and only considered random storage. Moreover, in a tier-to-tier AVS/RS, empty shuttles are always pooled at the ground level, and when conducting a retrieval task, a shuttle has to ride on a lift to travel to the target tier and go back to ground level with the requested SKU. Therefore, shuttles are not moved from one tier to another in an AVS/RS. Roy et al. (2012) considered the rules for moving shuttles between aisles on the same tier. For a given task, they classified shuttles into different classes based on the dwell point of available shuttles. However, instead of showing how shuttles are distributed over tiers, they only calculated the number of shuttles in each class and did not consider the interaction between the lift and shuttles, which can affect the performance of a tier-to-tier SBS/RS.

Compared with AVS/RS, SBS/RS received limited attention. Several studies have been done on different system configurations (Tappia et al., 2017; Wu et al., 2020; Xu et al., 2015) and the task sequencing problem (Carlo and Vis, 2012; Wang et al., 2015; Zhan et al., 2020) in tier-captive systems. The performance of a tier-captive SBS/RS with random storage policy was first measured by Lerher et al. (2015b) through simulation. They extended their study by developing expected travel time models for single-deep and double-deep tier-captive systems, respectively (Lerher et al., 2015a; Lerher, 2016). However, they modeled lifts and shuttles separately and ignored the possible waiting time for lifts to serve a requested SKU. Eder and Kartnig (2016) examined the throughput capacity of a tier-captive SBS/RS under random storage by simulation and provided insights on the ideal geometry of racks. Ekren et al. (2018) developed a tool to measure the expected travel time and energy consumption of a tier-captive SBS/RS with random storage. However, they followed the same method adopted by Lerher (2016) and therefore shared the same limitations. Recently, Eder (2019) modeled a tier-captive system as an open queuing

network and captured the interaction between lifts and shuttles. He later extended the study to a multi-deep tier-captive SBS/RS (Eder, 2020). Ekren and Akpunar (2021) developed a tool based on an open queuing network to estimate different performance metrics (e.g., average transaction time, energy consumption) for a single-deep tier-captive SBS/RS with random storage. Lerher et al. (2021) analyzed the performance for an SBS/RS where a tier-captive vehicle/crane can take care of the tasks from multi-tiers. However, even though a vehicle can reach multi-tiers, it does not need to travel across tiers. Therefore, the studied system is a tier-captive SBS/RS and does not need to consider shuttle relocation. As abovementioned, most of the researches on tier-captive systems assume random storage, and the studies on the impacts of alternative storage policies on the tier-captive systems' performance are limited. Ekren et al. (2015) considered the design of a tier-captive SBS/RS under class-based storage policy through simulation. Eder (2020) also estimated the performance of a tier-captive SBS/RS with class-based storage through an open queueing network. Kriehn et al. (2019) provided analytical model and algorithm for optimal tier-captive SBS/RS rack design under class-based storage. However, all of these studies focused on tier-captive SBS/RS and therefore did not consider the reallocation of shuttles.

Ha and Chae (2018) proposed strategies to avoid shuttle collision and policies to store incoming identical SKUs to reduce transaction time for a tier-to-tier SBS/RS. The efficiency of the proposed strategies was tested through simulation. However, they did not study the effects of storage policies and shuttle dispatching rules on system performance. They later developed a travel time model for a tier-to-tier SBS/RS with two lifts for each aisle, one for retrieval tasks and the other for storage tasks (Ha and Chae, 2019). By solving the travel time model sequentially regarding the number of shuttles, they determined the minimum number of shuttles in a given SBS/RS to meet a given throughput capacity requirement. However, they only considered random storage and random shuttle dispatching. Moreover, their travel time model was developed through the same method used by Lerher et al. (2015a) and therefore ignored the possible waiting time of a lift for an SKU. Our study will show that waiting time is an important component of travel time. Zhao et al. (2020) analyzed the expected travel time of a tier-to-tier SBS/RS with random storage. However, they only considered retrieval tasks and assumed that all spare shuttles are pooled at the ground level and therefore are no transferred across tiers. Küçükyaşar et al. (2021) compared the performance of tier-captive and tier-to-tier SBS/RS under random storage by simulation. However, they did not investigate the impact of the shuttle dispatching rules. Moreover, their

simulation model requires more running time than expected travel time models to compare multiple design choices and cannot facilitate efficient decision-making. Kriehn et al. (2018) investigated the impacts of class-based storage policy, tasks sequencing, and storage reorganization on the performance of a tier-to-tier SBS/RS but did not consider the impact of shuttle dispatching rules.

In summary, even though there are some studies to evaluate the performance of tier-to-tier SBS/RS under different storage policies, none of the exiting studies have paid attention to shuttle dispatching rules. However, the storage policy and shuttle dispatching rule combination are important in travel time modeling to evaluate system performance. Therefore, this study developed the expected travel time models for a tier-to-tier SBS/RS considering different storage policies, shuttle dispatching rules, and operational modes (i.e., SC and DC cycles) to bridge the research gap and facilitate the design and operations of tier-to-tier SBS/RS under different application environments.

3.4 Time Models Considering Retrieval Tasks and an SC cycle

Given the fact that many industries (e.g., online sales) require high responsiveness on retrieval tasks and may conduct retrieval tasks and storage tasks in different time windows, this study first considers the expected SC travel time for finishing a retrieval task in a tier-to-tier SBS/RS that allows fewer shuttles than tiers and then extends the results to DC operations. With the common assumptions that each aisle in the system is equipped with a lift and shuttles are aisle-captive (Eder, 2019; Ekren and Akpunar, 2021; Küçükyaşar et al., 2021; Lerher et al., 2021; Wang et al., 2015; Zhan et al., 2020; Zhao et al., 2019), we consider a one-aisle (two storage racks) tier-to-tier SBS/RS as an independent system. The assumption of aisle-captive shuttles is common because "specialized"/aisle-captive shuttles, which can only move along one direction, are usually used in practice. For traveling across aisles, "specialized" shuttles need to ride a transfer car or be replaced by "generic" shuttles, which can travel along different directions and are much more expensive than "specialized"/aisle-captive shuttles. Even when shuttles can travel across aisles, our developed expected travel time models will still be applicable if there is only one lift equipped for the rack system. When there are multiple lifts, the pairing between lifts and shuttles will significantly increase modeling difficulty.

In SC operations for retrieval tasks only, the lift has to handle SKU retrieval and possibly shuttle reallocation tasks. The lift or a shuttle remains at the position where it finishes its last job before performing a new task. Since only retrieval tasks are considered, the lift will remain at the ground level when there is no retrieval task; and each shuttle will stop at the buffer location at its tier when they are idle. Moreover, each tier can have at most one shuttle to avoid collision (Ha and Chae, 2019; Wu et al., 2020). The time for shuttles/lift to load/unload a task is ignored, which is also a common assumption in the literature (Dong et al., 2021; Fukunari and Malmborg, 2009; Kuo et al., 2007; Liu et al., 2020; Tappia et al., 2017). Similar to Eder (2020), we divide the rack into classes along vertical direction. To simplify the study, each tier is considered a class when the class-based storage policy is implemented. Therefore, all SKUs stored on one tier belong to one class and are randomly assigned to storage cells on the tier. Please note that the proposed models are also applicable to the situation in which the number of product classes are smaller than the number of tiers by giving multiple tiers the same demand rate. It is worth to note that many other studies considering class-based storage in SBS/RS, being consistent with studies of crane-based AS/RS, divide the rack into classes along horizontal direction (i.e., distance to the lift or buffer locations) to reduce the shuttles' travel time or lift's potential waiting time for SKU. However, the storage locations classification strategy proposed by this study (i.e., along vertical direction) aims to reduce the lift's travel time (i.e., SKU's potential waiting time for the lift). We assume the efficiency of each storage locations classification is affected by equipment's speed profile, number of equipment, and demand pattern. A comprehensive study is worthwhile in the future to compare these two classification strategies under different situations.

Consider a tier-to-tier SBS/RS with N tiers. All tiers are ranked in the order of their distances to ground level, i.e. tier 1 is on the ground level and tier N is at the top. C is the number of columns in an aisle, which represents the number of storage cells on a tier. The storage capacity of the system is $N \times C$. There are M shuttles, where $1 \leq M \leq N$. Let t_{oi} and t_{ij} denote the time for the lift traveling from the ground level to tier i and time from tier i to j, respectively. Since the SKUs on one tier are randomly stored, the expected travel time for a shuttle to finish a retrieval/storage task and the average demand rate of SKUs at tier/class i are expressed by t^r and d_i, respectively. Please note that t^r follows the same distribution across tiers. Without loss of generality, the demand rate from each tier are normalized, i.e. $\sum_{i=1}^{N} d_i = 1$. Random storage means $d_1 = \cdots = d_N$ while class-based storage implies $d_1 \geq \cdots \geq d_N$.

Let x_i be the probability for tier i having a shuttle when a retrieval task arrives. Two scenarios must be considered for developing the SC travel time model.

- If a task is for an SKU at tier i and there is a shuttle at the tier, the lift will start to travel to the target tier, and the shuttle will simultaneously travel to pick up the requested SKU and transport it to the buffer location at the tier. If the SKU is already at the buffer location when the lift arrives ($t_{oi} > t^r$), the lift will pick up the SKU and travel back to the I/O station. Otherwise (i.e., $t_{oi} < t^r$), the lift will have to wait until the shuttle moves the SKU to the buffer location. In this scenario, the expected travel time can be expressed as $\max(t_{oi}, t^r) + t_{oi}$.
- If a task is for an SKU at tier i without a shuttle, the lift has to travel to another tier j that has a shuttle, move the shuttle from tier j to tier i, wait at tier i until that shuttle hands over the SKU, and bring the required SKU to ground level. The travel time under this scenario is $t_{oj} + t_{ij} + t^r + t_{oi}$. The shuttle dispatching rules decide which shuttle should be moved to tier i.

For convenience, we define $a_i = \max(t_{oi}, t^r) + t_{oi}$, $b_i = t_{oi} + t^r$ and $c_{ij} = t_{ij} + t_{oj}$. Therefore, the expected SC travel time only considering retrieval tasks can be expressed as

$$ESC = \sum_{i=1}^{N} d_i x_i a_i + \sum_{i=1}^{N} d_i (1 - x_i)\left[b_i + \sum_{j \neq i} x_{j|i} p_i(j) c_{ij}\right], \quad (3.1)$$

where $x_{j|i}$ and $p_i(j)$ denote the conditional probability of tier j having a shuttle given that tier i does not have one and the probability that the shuttle at tier j, if there is one, is selected for tier i, respectively. The values of x_i, $x_{j|i}$ and $p_i(j)$ are affected by both the shuttle dispatching rule and storage policy. Please note that, in practice, a requested SKU might be blocked by the previous tasks from the same tier if they have not been picked up by the lift and are waiting in the buffer area. In that case, a transaction might be delayed. However, similar to many other travel time assessment studies (Lerher, 2016; Yang et al., 2017; Zaerpour et al., 2013), such delay is considered small and not captured by the expected travel time model expressed by Equation (3.1).

To obtain the shuttle distribution in the tier-to-tier SBS/RS under different shuttle dispatching rules and storage policies, we model the system as a DTMC by defining system states based on the location of shuttles when a task arrives. Please note that the system is a CTMC if we also

want to track transition time. However, since this study focuses on the distribution of shuttles, we only model the embedded DTMC. A system state s is defined by the set of tiers with shuttles, where $|s| = M$ and $s \subseteq \{1, \dots, N\}$. For example, $s = \{1,2, \dots, M\}$ means a state that all shuttles are located on the first M tiers when an order arrives. The state space is S and $|S| = \binom{N}{M}$. Let π_s be the steady-state probability of any state $s \in S$, which can be obtained by solving equations (3.2) and (3.3),

$$\pi P = \pi, \text{and} \tag{3.2}$$

$$\pi e = 1. \tag{3.3}$$

where $\pi = (\pi_s | s \in S)$ is the stationary probability vector, e is the unit vector, and P is the transition matrix determined by shuttle dispatching rules and storage policies. The probability that tier i has a shuttle can be represented as

$$x_i = \sum_{s \in S | i \in s} \pi_s, \qquad \forall\, i = 1, \dots, N. \tag{3.4}$$

3.4.1 Expected SC Travel Time Model under Random Shuttle Dispatching Rule

Under the random shuttle dispatching rule ($\mathcal{R}_1$), all shuttles have the same probability to be selected whenever a shuttle is required. For state s and tier $j \notin s$, a state set $\mathcal{S}_{s,j} = \{\forall s' \in S | s \cup s' = s \cup \{j\}\}$ is defined. In other words, $\mathcal{S}_{s,j}$ is the set of states in which all other shuttles are at the same tiers as s except one shuttle is at tier j, which does not belong to s. Please note that $|\mathcal{S}_{s,j}| = M$, which will be used in the proof to Theorem 3.1. We also define another state set $\bar{\mathcal{S}}_s = \{s' \in S; |s \cap s'| < M - 1\}$, which includes all states in which two or more shuttles are at different tiers from state s. It is obvious that $\{s\} \cup \bar{\mathcal{S}}_s \cup_{j \notin s} \mathcal{S}_{s,j} = S, \forall s \in S$. Assuming the current state is s, we have the following one-step transition probabilities.

1. The transition probability that the system will remain at state s is $P_{s,s} = \sum_{i \in s} d_i$, corresponding to the scenario in which the requested SKU is at a tier currently having a shuttle.
2. When a new retrieval task is at a tier $j \notin s$ (i.e., there is no shuttle at tier j right now), the tier receives a randomly selected shuttle. The system will switch into a state $s' \in \mathcal{S}_{s,j}$ with the transition probability of $P_{s,s'} = \frac{d_j}{M}$.

3. For any state $s' \in \bar{\mathcal{S}}_s$, we have $P_{s,s'} = 0$ because of $|s \cap s'| \geq 2$, implying that it is not possible to have two shuttle locations changed over one transition.

Theorem 3.1. *Under $\mathcal{R}_1$ with an SC cycle, $\frac{\pi_s}{\pi_{s'}} = \frac{\prod_{i\in s} d_i}{\prod_{i\in s'} d_i}$ holds for any two states $s, s' \in S$.*

Proof. As discussed above, for any $s \in S$, we can write equation (3.2) into

$$\pi_s = \pi_s \sum_{i\in s} d_i + \sum_{j\notin s} \sum_{s'\in \mathcal{S}_{s,j}} \pi_{s'} \frac{1}{M} d_{i|i\in s-s'\cap s} \qquad \forall\, s \in S. \quad (3.5)$$

Please note $d_{i|i\in s-s'\cap s}$ is the probability that a new task requests an SKU at a tier that belongs to s but does not belong to s'. When it happens, if a shuttle at tier j that does not belong to s is selected to handle the task, with the probability of $\frac{1}{M}$, the system will transit from s' to s. Please note that states s and s' have only a tier different (i.e., $|s - s' \cap s| = 1$). Plugging $\pi_{s'} = \frac{\prod_{i\in s'} d_i}{\prod_{i\in s} d_i}\pi_s$ into equation (3.5), we have

$$\begin{aligned}
\pi_s &= \pi_s \sum_{i\in s} d_i + \sum_{j\notin s} \sum_{s'\in \mathcal{S}_{s,j}} \frac{\prod_{i\in s'} d_i}{\prod_{i\in s} d_i} \pi_s \frac{1}{M} d_{i|i\in s-s'\cap s} \\
&= \pi_s \sum_{i\in s} d_i + \frac{\pi_s}{M} \sum_{j\notin s} \sum_{s'\in \mathcal{S}_{s,j}} \frac{(\prod_{i\in s'} d_i)\left(d_{i|i\in s-s'\cap s}\right)}{\prod_{i\in s} d_i} \\
&= \pi_s \sum_{i\in s} d_i + \frac{\pi_s}{M} \sum_{j\notin s} \sum_{s'\in \mathcal{S}_{s,j}} \frac{d_j \prod_{i\in s} d_i}{\prod_{i\in s} d_i} = \pi_s \sum_{i\in s} d_i + \frac{\pi_s}{M} \sum_{j\notin s} \sum_{s'\in \mathcal{S}_{s,j}} d_j \\
&= \pi_s \sum_{i\in s} d_i + \pi_s \sum_{j\notin s} d_j = \pi_s.
\end{aligned}$$

Therefore, we prove that $\pi_{s'} = \frac{\prod_{i\in s'} d_i}{\prod_{i\in s} d_i}\pi_s$ satisfies equation (3.2) and Theorem 3.1. ■

By applying Theorem 3.1 to the normalization equation (3.3), we can obtain the values of π_s and then the values of x_i. The expected SC travel time under the random shuttle dispatching rule ($ESC_{\mathcal{R}_1}$) can be calculated by equation (3.6).

$$ESC_{\mathcal{R}_1} = \sum_{i=1}^{N} \left(d_i x_i a_i + d_i (1 - x_i) \left(b_i + \sum_{j=1}^{N} \frac{x_{j|i} c_{ij}}{M} \right) \right). \quad (3.6)$$

Here, the values of $x_{j|i}$ can be obtained by solving a system with tiers $1, \dots, i-1, i+1, \dots, N$ from the original system and M shuttles. The demand rate on each tier is the same as it is in the original system. Here, we know that $p_i(j) = \frac{1}{M}$ because each of the M shuttles has the same probability of being selected for tier i due to the random shuttle dispatching rule.

3.4.2 Expected SC Travel Time Model under Distance-based Shuttle Dispatching Rule

When the current state is s and a new retrieval task requires an SKU at tier $i \notin s$, the distance-based shuttle dispatching ($\mathcal{R}_2$) rule will move the shuttle from tier j^* to tier i to perform this new task, where

$$j^* = \begin{cases} \max\limits_{j \in s, j<i} j & s \cap \{1,2,\dots,i-1\} \neq \emptyset \\ \min\limits_{j \in s} j & \text{Otherwise.} \end{cases} \quad (3.7)$$

In other words, the $\mathcal{R}_2$ rule moves the closest shuttle from a tier lower than tier i, if there is any. Otherwise, the rule moves the closest shuttle from a higher tier. Please note that the lift always starts from the ground level in an SC cycle, so there is an incentive to move a shuttle from a lower tier rather than from a higher tier to reduce the travel distance of the lift moving a shuttle to tier *i*. This incentive motivates the $\mathcal{R}_2$ rule. The transition probabilities P from state s through a new task at tier i are as follows.

1. $P_{s,s} = \sum_{i \in s} d_i$, which is for the case that $i \in s$.
2. If $i \notin s$, $P_{s,s'} = d_i$, where $s' = s/\{j^*\} \cup \{i\}$ and j^* is defined in (3.7).
3. For any other states s', $P_{s,s'} = 0$.

Theorem 3.2*. Under $\mathcal{R}_2$ with an SC cycle, the steady-state probability distribution and the probability for tier i to have a shuttle are as follows.*

$$\pi_s = \begin{cases} \frac{d_i}{\sum_1^{N-M+1} d_k} & s = \{i, N-M+2, N-M+3, \dots, N\}, \forall i \in \{1,2,\dots,N-M+1\}, \\ 0 & Otherwise. \end{cases} \quad (3.8)$$

and

$$x_i = \begin{cases} 1 & i \in \{N-M+2, N-M+3, \dots, N\}, \\ \frac{d_i}{\sum_1^{N-M+1} d_k} & i \in \{1,2, \dots, N-M+1\}. \end{cases} \quad (3.9)$$

Proof. Define state sets $\mathbb{S}_1 = \{s = \{i, N-M+2, N-M+3, \dots, N\}, \forall i \in \{1,2, \dots, N-M+1\}\}$ and $\mathbb{S}_2 = S/\mathbb{S}_1$. In other words, for any state belonging to $\mathbb{S}_1$, all shuttles except one stay at top $M-1$ tiers and the remaining one is at one of the $N-M+1$ tiers. It is obvious that there is no path from $\mathbb{S}_1$ to $\mathbb{S}_2$ but there is a path from any state s' in $\mathbb{S}_2$ to a state s in $\mathbb{S}_1$. Therefore, $\pi_s = 0, \forall s \in \mathbb{S}_2$. All states in $\mathbb{S}_1$ communicate with each other and there are no cycles. The transition probability to a state $s = \{i, N-M+2, N-M+3, \dots, N\}, \forall i \in \{1,2, \dots, N-M+1\}$ in $\mathbb{S}_1$ from each of other states in $\mathbb{S}_1$ is d_i, which leads to (3.8) and then (3.9) for x_i under $\mathcal{R}_2$. ■

Theorem 3.2 means that all shuttles except one stay at top tiers and one shuttle travels back and forth among all $N-M+1$ tiers from the bottom in the long-run. Therefore, $p_i(j) = \begin{cases} 1 & j < N-M+1, j \neq i \\ 0 & j \geq N-M+1 \end{cases}$ for $i \in \{1, \dots, N-M+1\}$. The expected SC travel time under $\mathcal{R}_2$, $ESC_{\mathcal{R}2}$, is

$$ESC_{\mathcal{R}_2} = \sum_{i=N-M+2}^{N} d_i a_i + \sum_{i=1}^{N-M+1} \frac{d_i^2 a_i + b_i d_i (D - d_i) + \sum_{j=1,\dots,N-M+1; j \neq i} d_i d_j c_{ij}}{D}, \text{where} \quad (3.10)$$

$$D = \sum_{k=1}^{N-M+1} d_k. \quad (3.11)$$

Theorem 3.3*. Under random storage and an SC cycle, $ESC_{\mathcal{R}_1} - ESC_{\mathcal{R}_2} \geq 0$ and the difference is concave in terms of M.*

The proof to Theorem 3.3 is provided in Appendix F. For better understanding the expected SC travel time, $ESC_{\mathcal{R}}$ is partitioned into three time components: expected time for the lift to travel between tiers for retrieving an SKU ($TSC_r^{\mathcal{R}}$), expected time for moving a shuttle ($TSC_m^{\mathcal{R}}$), and expected time waiting for the shuttle to move the SKU ($TSC_w^{\mathcal{R}}$), where $\mathcal{R} \in \{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3\}$ represents the shuttle dispatching rule. These three terms can be represented by equation (3.12-

3.14), where $x_i, x_{j|i}$ and $p_i(j)$ are related to $\mathcal{R}$. Here, $TSC_r^{\mathcal{R}}$ is constant once the storage policy is fixed. Those components are used in Appendix F to prove Theorem 3.3 and in numerical result discussion.

$$TSC_r^{\mathcal{R}} = \sum_{i=1}^{N} 2d_i t_{oi}, \tag{3.12}$$

$$TSC_m^{\mathcal{R}} = \sum_{i=1}^{N} d_i(1-x_i)\left[\sum_{j=1}^{N} x_{j|i} p_i(j) c_{ij} - t_{oi}\right], \text{and} \tag{3.13}$$

$$TSC_w^{\mathcal{R}} = \sum_{i=1}^{N} d_i x_i [t^r - t_{oi}]^+ + d_i(1-x_i)t^r, \tag{3.14}$$

3.4.3 Expected SC Travel Time Model under Demand-rate-based Shuttle Dispatching Rule

To reduce the frequency of shuttle movements, the demand-rate-based shuttle dispatching rule ($\mathcal{R}_3$) selects the shuttle from the tier with the lowest demand rate whenever a shuttle is needed. Under $\mathcal{R}_3$, we number tiers based on their demand rates rather than locations with a descending order. In other words, $d_{[1]} \geq d_{[2]}, \dots, d_{[N]}$. The transition probabilities P from state s to another state due to a new task at tier i are as follows. Please note that the index system $\{[1], \dots, [N]\}$ is the same as $\{1, \dots, N\}$ under class-based storage. Moreover, since $d_1 = d_2 = \cdots = d_N$ under random storage, $\mathcal{R}_3$ is the same as $\mathcal{R}_1$ when SKUs are randomly accommodated. The state transition probabilities are listed as follows.

1. $P_{s,s} = \sum_{[i] \in s} d_{[i]}$, which is for the case that $[i] \in s$.
2. If $[i] \notin s$, $P_{s,s'} = d_{[i]}$, where $s' = s \Big/ \left\{ \underset{[j] \in s}{\operatorname{argmin}}\, d_{[j]} \right\} \cup \{[i]\}$.
3. For any other states s', $P_{s,s'} = 0$.

Theorem 3.4*. Under $\mathcal{R}_3$ and an SC cycle, the steady-state probability distribution and the probability for tier i to have a shuttle are as follows.*

$$\pi_s = \begin{cases} \dfrac{d_{[i]}}{\sum_{k=1}^{M-1} d_{[k]}} & s = \{[1], [2], \dots, [M-1], [i]\}, \forall [i] \in \{M, M+1, \dots, N\}, \\ 0 & \text{Otherwise.} \end{cases} \tag{3.15}$$

and

$$x_{[i]} = \begin{cases} 1 & [i] \in \{1,2,\dots,M-1\}, \\ \dfrac{d_{[i]}}{\sum_{[k]=M}^{N} d_{[k]}} & [i] \in \{M, M+1, \dots, N\}. \end{cases} \tag{3.16}$$

The proof to Theorem 3.4 is very similar to the proof to Theorem 3.2. Under $\mathcal{R}_3$, all shuttles, except one, stay at the tiers with the $M-1$ highest demand, and the other shuttle travels among other tiers. Therefore,

$$p_{[i]}([j]) = \begin{cases} 1 & [j] \geq M, [j] \neq [i] \\ 0 & [j] < M \end{cases} \text{ for } [i] \geq M \text{ under } \mathcal{R}_3.$$

Under class-based storage, $M-1$ shuttles stay at the bottom tiers because they have the highest demand rates, and the other shuttle travels among tiers M to N. The expected travel time under the demand-based shuttle dispatching rule, $ESC_{\mathcal{R}_3}$, is

$$ESC_{\mathcal{R}_3} \tag{3.17}$$
$$= \sum_{[i]=1}^{M-1} d_{[i]} a_{[i]} + \sum_{[i]=M}^{N} \frac{d_{[i]}^2 a_{[i]} + d_{[i]}\left(\sum_{k=M}^{N} d_{[k]} - d_{[i]}\right) b_{[i]} + \sum_{[j]=M,\dots,N;[j]\neq i} d_{[i]} d_{[j]} c_{[i][j]}}{\sum_{[k]=M}^{N} d_{[k]}}.$$

3.5 Travel Time Models Considering Both Retrieval Tasks and Storage Tasks in DC Cycles

In section 3.4, the SC travel time models under different combinations of storage policies and shuttle dispatching rules are developed. Storage tasks could also be important and are conducted simultaneously with retrievals in many industries. This section will develop the DC travel time models that consider both retrieval and storage tasks. More specifically, we will consider the retrieval tasks and the re-storage of the retrieved SKUs. At each DC cycle, the lift first stores the SKU retrieved during the last retrieval operation to its storage location and then executes a retrieval task if there is one. Without loss of generality, we assume that the number of retrieval tasks equals the number of the re-storage tasks. In addition, the re-storage tier always has a shuttle.

The assumptions, notations, and methods adopted in section 3.4 are also applied for developing the DC travel time models. Let $x'_{j|i}$ be the probability for tier j having a shuttle at the beginning

of each DC cycle given the re-storage location is tier i (i.e., there is a shuttle at tier i). Two scenarios must be considered for developing the DC travel time models.

- If the re-storage location is tier i and the retrieval task is from tier j with a shuttle, the lift will start to travel to tier i from the I/O station to drop the SKU that needs to be stored and then travel to tier j. The shuttle on tier j will simultaneously travel to pick up the requested SKU and transport it to the buffer location at the tier. If the SKU is already at the buffer location when the lift arrives ($t_{oi} + t_{ij} > t^r$), the lift will pick up the SKU and travel back to the I/O station. Otherwise (i.e., $t_{oi} + t_{ij} < t^r$), the lift will have to wait until the shuttle moves the SKU to the buffer location. In this scenario, the expected DC travel time can be expressed as $\max(t_{oi} + t_{ij}, t^r) + t_{oj}$. Please note that when $i = j$ (i.e., re-storage and retrieval are from the same tier), the shuttle will conduct the retrieval task while the lift is traveling from the I/O station to that tier. After the retrieved SKU is picked up by the lift, the shuttle can execute the re-storage tasks. In that case, the expected DC travel time can be expressed as $\max(t_{oi}, t^r) + t_{oi}$.
- If the retrieval task is for an SKU at tier j without a shuttle, the lift, after dropping the re-stored SKU at tier i, has to travel to another tier k that has a shuttle, move the shuttle from tier k to tier j, wait at tier j until when that shuttle hands over the SKU, and bring the required SKU to the ground level. If $k = i$, then the DC travel time is $t_{oi} + t^r + t_{ij} + t^r + t_{oj}$, since the shuttle on tier i has to conduct the re-storage task before being relocated. Otherwise (i.e., $k \neq i$), the DC travel time is $t_{oi} + t_{ik} + t_{kj} + t^r + t_{oj}$. The shuttle dispatching rules decide which shuttle should be moved to tier j.

For convenience, we define $a'_{ij} = \max(t_{oi} + t_{ij}, t^r) + t_{oj}$ and

$$e_{ikj} = \begin{cases} t_{oi} + t^r + t_{ij} + t^r + t_{oj} & if\ k = i \\ t_{oi} + t_{ik} + t_{kj} + t^r + t_{oj} & \text{Otherwise.} \end{cases}$$

The expected DC travel time is

$$EDC = \sum_{i=1}^{N}\sum_{j=1}^{N} d_i d_j\, x'_{j|i} a'_{ij} + \sum_{i=1}^{N}\sum_{j=1}^{N} d_i d_j\,(1 - x'_{j|i}) \sum_{k=1}^{N} x_{k|(i,j)} p_{(i,j)}(k) e_{ikj}\,, \qquad (3.18)$$

where $x_{k|(i,j)}$ and $p_{(i,j)}(k)$ denote the conditional probability of tier k having a shuttle given that tier i has one and tier j does not have one, and the probability that the shuttle at tier k, if there is one, is selected for tier j, respectively. The values of $x'_{j|i}$, $x_{k|(i,j)}$ and $p_{(i,j)}(k)$ are affected by both the shuttle dispatching rule and storage policy.

3.5.1 Expected DC Travel Time Model under Random Shuttle Dispatching Rule

The random shuttle dispatching rule ($\mathcal{R}_1$) for a DC cycle remains the same as that under an SC cycle. All shuttles, including the shuttle at the tier for the re-storage task, have the same probability to be selected whenever a shuttle is requested. Therefore, under a DC cycle, Theorem 3.1 still holds when $\mathcal{R}_1$ is implemented. The expected DC travel time model under $\mathcal{R}_1$, $EDC_{\mathcal{R}_1}$, can be expressed as

$$EDC_{\mathcal{R}_1} = \sum_{i=1}^{N}\sum_{j=1}^{N} d_i d_j \, x'_{j|i} a'_{ij} + \sum_{i=1}^{N}\sum_{j=1}^{N} d_i d_j \, (1 - x'_{j|i}) \sum_{k=1}^{N} \frac{x_{k|(i,j)} e_{ikj}}{M}. \tag{3.19}$$

Here, $x'_{j|i} = 1$ when $i = j$, and when $i \neq j$, $x'_{j|i}$ can be obtained by solving a system with tiers $1, \dots, i-1, i+1, \dots, N$ from the original system and $M - 1$ shuttles. Solving a system with tiers $1, \dots, i-1, i+1, , \dots, j-1, j+1, \dots, N$ from the original system and $M - 1$ shuttles can give the values of $x_{k|(i,j)}$ when $k \notin \{i, j\}$, and $x_{i|(i,j)} = 1$ and $x_{j|(i,j)} = 0$. Here, we know that $p_{(i,j)}(k) = \frac{1}{M}$. because each of the M shuttles has the same probability of being selected for tier j due to the random shuttle dispatching rule.

3.5.2 Expected DC Travel Time Model under Distance-based Shuttle Dispatching Rule

For the tier-to-tier SBS/RS under DC operations, when the re-storage task is from tier i, the retrieval task is from tier j without a shuttle, and $i \neq j$, the distance-based shuttle dispatching ($\mathcal{R}_2$) rule, instead of selecting the shuttle closest to tier j, will always move the shuttle from tier i to tier j to minimize the lift's travel distance for shuttle reallocation.

Recall the state s, and state set $\mathcal{S}_{s,j}$ and $\bar{\mathcal{S}}_s$ defined in subsection 3.4.1, we have the following one-step transition probabilities by assuming the current state is s. The transition probability that the system will remain at state s is $P_{s,s} = \sum_{i \in s} d_i$, corresponding to the scenario that the retrieval task is at a tier currently having a shuttle.

1. When a new retrieval task is at a tier $j \notin s$ (i.e., there is no shuttle at tier j right now), the tier receives the shuttle from tier $i \in s$, where the re-storage task is located. The system will switch into a state $s' \in \mathcal{S}_{s,j}$ with the transition probability of $P_{s,s'} = \frac{d_i d_j}{\sum_{i \in s} d_i}$, where $i = s \cup s' - s \cup \{j\}$, and $\frac{d_i}{\sum_{i \in s} d_i}$ is the conditional probability that the re-storage task is at tier $i \in s$ given the fact that shuttles are located on tiers in s at the beginning of this cycle.
2. For any state $s' \in \bar{\mathcal{S}}_s$, we have $P_{s,s'} = 0$ because $|s \cap s'| \geq 2$, implying that it is not possible to have two shuttle locations changed over one transition.

The transition probabilities P from state s through a new retrieval task at tier j are as follows.

1. $P_{s,s} = \sum_{j \in s} d_j$, which is for the case that $j \in s$.
2. If $j \notin s$, $P_{s,s'} = \frac{d_i d_j}{\sum_{i \in s} d_i}$, where $s' = s/\{i\} \cup \{j\}$.
3. For any other states s', $P_{s,s'} = 0$.

Theorem 3.5*. Under R_2 and a DC cycle, for any two states $s, s' \in S$, $\frac{\pi_s}{\pi_{s'}} = \frac{\sum_{i \in s} d_i}{\sum_{i \in s'} d_i}$ holds.*

Proof. As discussed above, for any $s \in S$, we can write equation (3.2) into

$$\pi_s = \pi_s \sum_{i \in s} d_i + \sum_{j \notin s} \sum_{s' \in \mathcal{S}_{s,j}} \pi_{s'} \, d_{ij|i \in s - s' \cap s, j \in s' - s' \cap s} \qquad \forall \, s \in S. \qquad (3.20)$$

Please note that $d_{ij|i \in s - s' \cap s, j \in s' - s' \cap s}$ is the joint probability that a new retrieval task requests an SKU at tier i that belongs to s but does not belong to s' and the re-storage location is at tier j that belongs to s' but does not belong to s. Please also note that $d_{ij|i \in s - s' \cap s, j \in s' - s' \cap s} = \frac{d_j d_{i|i \in s - s' \cap s}}{\sum_{i \in s'} d_i}$, where $d_{i|i \in s - s' \cap s}$, as defined in subsection 3.4.1, is the probability that a new task requests an SKU at a tier that belongs to s but does not belong to s'. When it happens, with the probability of 1, the system will transit from s' to s, where $s = s'/\{j\} \cup \{i\}$. Plugging $\pi_{s'} = \frac{\sum_{i \in s'} d_i}{\sum_{i \in s} d_i} \pi_s$ into equation (3.5), we have

$$\pi_s = \pi_s \sum_{i\in s} d_i + \sum_{j\notin s} \sum_{s'\in \mathcal{S}_{s,j}} \frac{\sum_{i\in s'} d_i}{\sum_{i\in s} d_i} \pi_s \, d_{ij|i\in s-s'\cap s, j\in s'-s'\cap s}$$

$$= \pi_s \sum_{i\in s} d_i + \pi_s \sum_{j\notin s} \sum_{s'\in \mathcal{S}_{s,j}} \frac{\sum_{i\in s'} d_i}{\sum_{i\in s} d_i} \frac{d_j d_{i|i\in s-s'\cap s}}{\sum_{i\in s'} d_i}$$

$$= \pi_s \sum_{i\in s} d_i + \pi_s \sum_{j\notin s} \sum_{s'\in \mathcal{S}_{s,j}} \frac{d_j d_{i|i\in s-s'\cap s}}{\sum_{i\in s} d_i} = \pi_s \sum_{i\in s} d_i + \pi_s \sum_{j\notin s} d_j = \pi_s.$$

Therefore, we prove that $\pi_{s'} = \frac{\sum_{i\in s'} d_i}{\sum_{i\in s} d_i} \pi_s$ satisfies equation (3.2) and Theorem 3.5. ■

By applying Theorem 3.5 to the normalization equation (3.3), we can obtain the values of π_s and then the values of x_i and $x'_{j|i}$. The expected travel time under the distance-based shuttle dispatching rule ($EDC_{\mathcal{R}_2}$) in a DC cycle can be calculated by equation (3.21).

$$EDC_{\mathcal{R}_2} = \sum_{i=1}^{N} \sum_{j=1}^{N} d_i d_j \, x'_{j|i} a'_{ij} + \sum_{i=1}^{N} \sum_{j=1}^{N} d_i d_j \, (1 - x'_{j|i}) e_{iij}. \tag{3.21}$$

It is worth to note that Theorem 3.3 indicates that $\mathcal{R}_2$ is always better than $\mathcal{R}_1$ under SC operations and random storage. However, it is not true for DC operations. When random storage is adopted (i.e., $d_1 = d_2 = \cdots = d_N$),

$$EDC_{\mathcal{R}_2} - EDC_{\mathcal{R}_1}$$

$$= \frac{(N-M)}{N^2(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} e_{iij} - \frac{(N-M)}{N^2(N-1)M} \sum_{i=1}^{N} \sum_{j=1}^{N} \Big(\sum_{\substack{k=1 \\ k\neq i \\ k\neq j}}^{N} \frac{M-1}{N-2} e_{ikj} + e_{iij} \Big).$$

Because $e_{ikj} = \begin{cases} t_{oi} + t^r + t_{ij} + t^r + t_{oj} & if\ k = i \\ t_{oi} + t_{ik} + t_{kj} + t^r + t_{oj} & \text{Otherwise,} \end{cases}$

$$EDC_{\mathcal{R}_2} - EDC_{\mathcal{R}_1}$$

$$= \frac{(N-M)}{N^2(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} \left(t_{ij} + t^r\right)$$

$$- \frac{(N-M)}{N^2(N-1)M} \sum_{i=1}^{N} \sum_{j=1}^{N} \left(\sum_{\substack{k=1 \\ k \neq i \\ k \neq j}}^{N} \frac{M-1}{N-2} \left(t_{ik} + t_{jk}\right) + t_{ij} + t^r \right).$$

Therefore, $EDC_{\mathcal{R}_2} - EDC_{\mathcal{R}_1}$ is essentially affected by equipment speeds. When shuttles are fast enough, or the lift is very slow, $EDC_{\mathcal{R}_2} - EDC_{\mathcal{R}_1} < 0$; otherwise, $EDC_{\mathcal{R}_2} - EDC_{\mathcal{R}_1} > 0$.

3.5.3 Expected DC Travel Time Model under Demand-rate-based Shuttle Dispatching Rule

The demand-rate-based shuttle dispatching rule ($\mathcal{R}_3$) for DC operations also remains the same as it is under SC operations. $\mathcal{R}_3$ always selects the shuttle from the tier with the lowest demand rate whenever a shuttle is needed. Therefore, when $\mathcal{R}_3$ is implemented, Theorem 3.4 still holds under DC operations. The expected DC travel time model under $\mathcal{R}_3$, $EDC_{\mathcal{R}_3}$, can be expressed as

$$\begin{aligned} EDC_{\mathcal{R}_3} = & \sum_{[j]=1}^{N} \left(\sum_{[i]=1}^{M-1} d_{[i]} d_{[j]} a'_{[i][j]} + \sum_{[i]=\mathrm{M}}^{N} \frac{d_{[i]} d_{[j]} a'_{[i][j]} d_{[i]}}{\sum_{[k]=M}^{N} d_{[k]}} \right) \\ & + \sum_{[j]=1}^{M-1} \sum_{[i]=\mathrm{M}}^{N} d_{[i]} d_{[j]} \left(1 \right. \\ & \left. - \frac{d_{[i]}}{\sum_{[k]=M}^{N} d_{[k]}} \right) \sum_{\substack{[k]=\mathrm{M} \\ [k] \neq [i]}}^{N} \frac{d_{[k]}}{\sum_{[k]=M}^{N} d_{[k]} - d_{[i]}} e_{[j][k][i]} \\ & + \sum_{[j]=M}^{N} \sum_{[i]=\mathrm{M}}^{N} d_{[i]} d_{[j]} \left(1 - \frac{d_{[i]}}{\sum_{[k]=M}^{N} d_{[k]}} \right) e_{[j][j][i]}. \end{aligned} \tag{3.22}$$

3.6 Numerical Results and Discussion

The parameters of the tier-to-tier SBS/RS in numerical experiments were from Ha & Chae (2019). As discussed before, we only need to consider a one-aisle system with two racks. The system has $N = 12$ tiers and 15 columns on each rack, which can hold 360 SKUs. The height and

width (along the aisle) of one storage cell are $0.5\ m$ and $0.6\ m$, respectively. The length of one storage cell is not important since it does not affect the equipment's travel times. The maximum velocity of shuttles is $V_s = 1.5\ m/s$, and their acceleration/deceleration is $a_s = 1.5\ m/s^2$. The maximum velocity of the lift is $v_l = 1.5\ m/s$, and its acceleration/deceleration is $a_l = 1.5\ m/s^2$. For class-based storage, we used the method proposed by Hausman et al. (1976) to calculate the average demand rate for SKUs on each tier. They modeled the well-known 'ABC' phenomenon for inventory as

$$G(l) = l^u, \qquad 0 < l \leq 1 \tag{3.23}$$

where $0 < u \leq 1$. Formulation (3.23) represents the ranked cumulative $(l^u \times 100)\%$ demand versus $(l \times 100)\%$ of stored SKUs. The value of u determines the distribution of demand. A greater u means more homogenous demand rates across SKUs. For a given u, d_i is obtained through equations (3.24) and (3.25) (Hausman et al. 1976).

$$d_i = G\left(\frac{i}{N}\right) - G\left(\frac{i-1}{N}\right) \qquad i = 1, \dots, N \tag{3.24}$$

$$G(0) = 0 \tag{3.25}$$

We consider nine demand distribution scenarios, which is featured by $G(0.1) \in \{0.1, 0.2, 0.3, \dots, 0.9\}$, for numerical experiments. Here, $G(0.1)$ means the percentage of demand of the top 10% SKUs, which is at least 10%. For example, $u = 0.69897$ for the scenario of $G(0.1) = 0.2$ can be obtained by solving equation (3.23) of $(0.1)^u = 0.2$ and means that the top 10% SKUs have 20% of the total demand. $G(0.1) = 0.1$ (i.e., $u = 1$) implies that all SKUs have the same demand rates and class-based storage is the same as random storage. Since the $\mathcal{R}_3$ shuttle dispatching rule is not designed for random storage, we only apply $\mathcal{R}_1$ and $\mathcal{R}_2$ for random storage.

3.6.1 Travel Time Models Validation through Simulation

For each of the nine demand distribution scenarios, we randomly created 10 task sequences, where the number of retrieval tasks is uniformly distributed between 20 and 360. For numerical experiments of DC operations, the re-storage tasks in the m^{th} cycle corresponds to the retrieval task in the $(m-1)^{th}$ cycle. We perturbed the number of shuttles in the system with $M = 1, 2, \dots, 12$. We define $|\Delta|$ as the average absolute difference between simulation results and travel times calculated by developed models over 10 random task sequences under a given

storage policy, shuttle dispatching rule, demand distribution, number of shuttles, and operational mode (i.e., SC or DC). Furthermore, σ represents the standard deviation of the differences across the 10 instances. Table 3-1 lists both $|\Delta|$ and σ (in seconds) for the demand distribution scenario of $G(0.1) = 0.5$ under SC operations, and shows all $|\Delta|$ are much smaller than σ. Actually, simulations for all scenarios under both SC and DC cycle yielded $|\Delta| \ll \sigma$, validating the accuracy of the developed travel time models.

3.6.2 Shuttle Dispatching Rule Comparison

3.6.2.1 Shuttle Dispatching Rule Comparison under SC Operations

Numerical experiments were first conducted to compare the two storage policies and three shuttle dispatching rules by only considering retrieval tasks under SC operations. Figure 3-2 shows the expected SC travel time of the proposed shuttle dispatching rules under different demand distributions and the number of shuttles, M. The best dispatching rule under each scenario is selected, and the advantage against the second-best dispatching rule is calculated through $\frac{ESC_B - ESC_{SB}}{ESC_{SB}} \times 100\%$ and listed in each cell, where ESC_B and ESC_{SB} denote the expected SC travel time under the best shuttle dispatching rule and the second-best one, respectively. Here, green means that $\mathcal{R}_2$ is the best while red means that $\mathcal{R}_3$ is the best.

The whole Figure 3-2 shows the following observations when class-based storage is used and only retrieval tasks are considered in SC operations.

- The random shuttle dispatching rule $\mathcal{R}_1$ is dominated. The best dispatching rule is either $\mathcal{R}_2$ or $\mathcal{R}_3$, depending on the demand distribution and the number of shuttles M. The difference of travel times between the best and second-best dispatching rules could be as high as 10.8%, which justifies the need of this study on shuttle dispatching rules.
- When the demand rate is rather homogenous (e.g., $G(0.1) = 0.1$), the distance-based shuttle dispatching rule $\mathcal{R}_2$ is in general better than the demand-rate-based shuttle dispatching rule $\mathcal{R}_3$.
- When the demand rate is a little more heterogeneous (e.g., $G(0.1) = 0.2$ or 0.3), $\mathcal{R}_3$ is the best when the number of shuttles is low, while more shuttles make $\mathcal{R}_2$ more competitive.
- When the demand rate is even more heterogeneous (e.g., $G(0.1) \geq 0.4$), $\mathcal{R}_3$ is dominant, but the advantage of $\mathcal{R}_3$ over the second-best rule increases and then decreases in the

Table 3-1. Comparison between Simulation and Expected SC Travel Time Models with $G(0.1) = 0.5$

	M	1	2	3	4	5	6	7	8	9	10	11	12
$\mathcal{R}_1$	$\lvert\Delta\rvert$	0.00	0.11	0.18	0.01	0.17	0.02	0.02	0.11	0.02	0.07	0.02	0.09
	σ	0.77	0.33	0.50	0.51	0.59	0.35	0.48	0.95	0.41	0.31	0.47	0.24
$\mathcal{R}_2$	$\lvert\Delta\rvert$	0.00	0.01	0.02	0.12	0.04	0.03	0.08	0.03	0.03	0.00	0.02	0.09
	σ	0.77	0.34	0.47	0.63	0.39	0.30	0.29	0.55	0.24	0.22	0.46	0.24
$\mathcal{R}_3$	$\lvert\Delta\rvert$	0.00	0.09	0.16	0.05	0.08	0.01	0.00	0.22	0.02	0.06	0.06	0.09
	σ	0.77	0.38	0.53	0.58	0.52	0.39	0.35	0.97	0.35	0.27	0.53	0.24

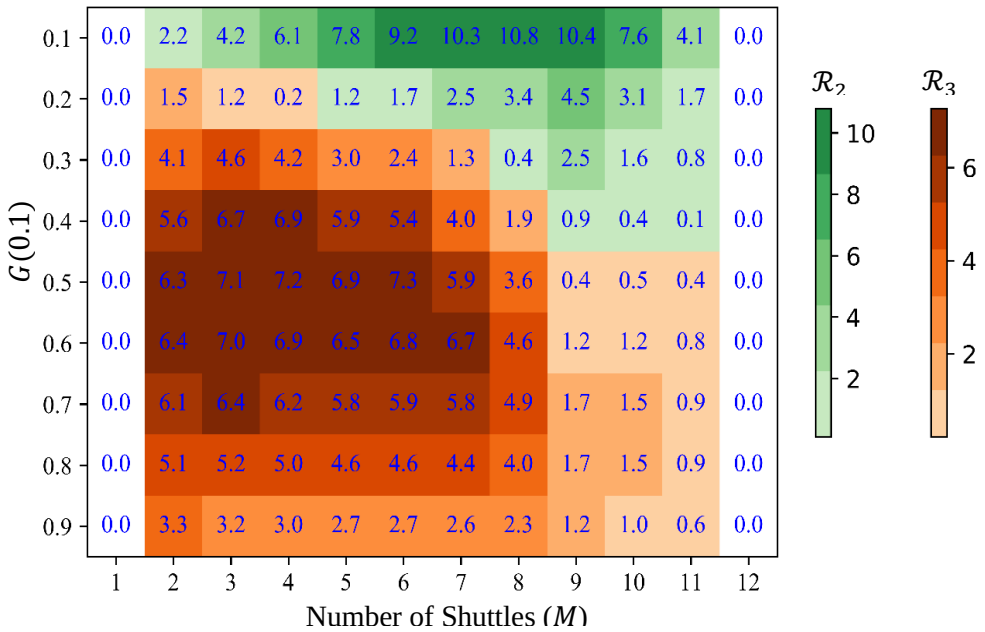
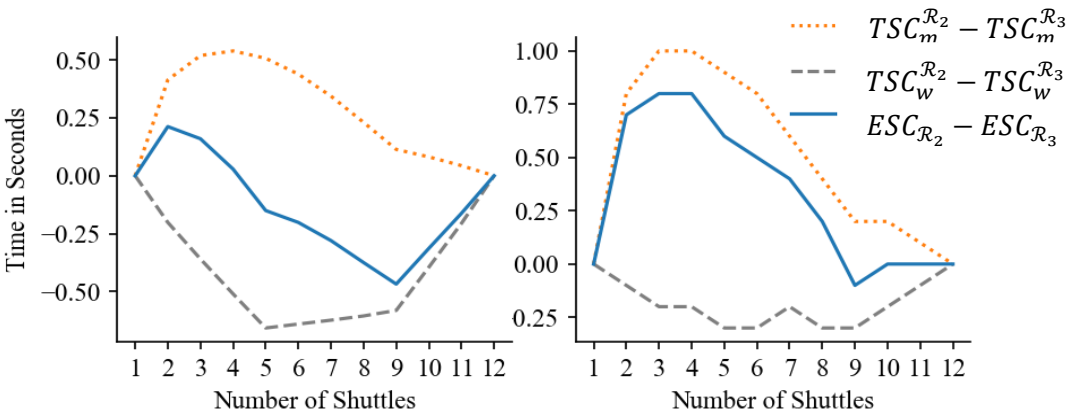


Figure 3-2. Best Dispatching Rules under Various $G(0.1)$ and M Values under Class-based Storage and SC Operations

number of shuttles. When the demand rate is highly heterogeneous (e.g., $G(0.1) = 0.9$), the advantage of $\mathcal{R}_3$ becomes smaller because shuttles are not moved often, especially when there are many shuttles.

- The green $G(0.1) = 0.1$ row validates Theorem 3.3 that $EC_{\mathcal{R}_1} - EC_{\mathcal{R}_2} \geq 0$ under random storage and shows the difference is concave in terms of M.

The observations from Figure 3-2 are consistent with those from Figure 3-3, which compares $\mathcal{R}_2$ and $\mathcal{R}_3$ when $G(0.1) = 0.2$ or $G(0.1) = 0.4$. Since $TSC_r^{\mathcal{R}}$ defined by (3.12) is constant once storage policy and demand pattern are given, only $ESC_{\mathcal{R}_2} - ESC_{\mathcal{R}_3}$, $TSC_m^{\mathcal{R}_2} - TSC_m^{\mathcal{R}_3}$ and $TSC_w^{\mathcal{R}_2} - TSC_w^{\mathcal{R}_3}$ are plotted in Figure 3-3. As discussed in subsection 3.4.2, a tier-to-tier SBS/RS under $\mathcal{R}_2$ can be separated into two subsystems: (1) a tier-captive SBS/RS formed by the top $M-1$ tiers with $M-1$ shuttles, and (2) a tier-to-tier SBS/RS consisting of the first $N-M+1$ tiers from the ground level with one shuttle. Similarly, $\mathcal{R}_3$ also separates the SBS/RS into two subsystems: (1) a tier-captive SBS/RS formed by the bottom $M-1$ tiers and (2) a tier-to-tier SBS/RS composed by the top $N-M+1$ tiers with one shuttle. To further explain why a dispatching rule is the best under different situations, we define $p_m^{\mathcal{R}}$ as the probability of needing shuttle reallocation for a task and $t_{m,i}^{\mathcal{R}}$ as the expected movement distance for reallocating a shuttle to tier i under shuttle dispatching rule $\mathcal{R}$. It is clear that $TSC_m^{\mathcal{R}}$, the expected shuttle movement distance (or time) per task, is influenced by both $p_m^{\mathcal{R}}$ and $t_{m,i}^{\mathcal{R}}$. Under class-based storage (i.e., lower tiers having higher demand rates), $p_m^{\mathcal{R}_3} \leq p_m^{\mathcal{R}_2}$, and $p_m^{\mathcal{R}_2} - p_m^{\mathcal{R}_3}$ increases and then decreases in M. Moreover, $\sum_{i=1}^{N} t_{m,i}^{\mathcal{R}_3} \approx \sum_{i=1}^{N} t_{m,i}^{\mathcal{R}_2}$, and therefore $TSC_m^{\mathcal{R}_2} - TSC_m^{\mathcal{R}_3} \geq 0$ and the difference increases and then decreases in M. Additionally, even though $\mathcal{R}_2$ yields lower lift waiting time $TSC_w^{\mathcal{R}_2}$, the difference from $TSC_w^{\mathcal{R}_3}$ becomes smaller while more demands aggregate on lower tiers (i.e., higher value of $G(0.1)$). Furthermore, $TSC_w^{\mathcal{R}_3} - TSC_w^{\mathcal{R}_2} > 0$ when $G(0.1)$ is small, but $TSC_w^{\mathcal{R}_2}$ is not always less than $TSC_w^{\mathcal{R}_3}$. $TSC_w^{\mathcal{R}_2} - TSC_w^{\mathcal{R}_3} > 0$ can be observed when $G(0.1)$ and M are extremely high. However, $TSC_w^{\mathcal{R}_2} - TSC_w^{\mathcal{R}_3}$ is dominated by $TSC_m^{\mathcal{R}_2} - TSC_m^{\mathcal{R}_3}$ when $G(0.1)$ is not too small. Due to the relationship of $t_{o3} < t^r < t_{o4}$ in the numerical experiments, the potential impact of shuttle dispatching rule on $TSC_w^{\mathcal{R}}$ is limited. A bigger



a. $G(0.1) = 0.2$ b. $G(0.1) = 0.4$

Figure 3-3. Comparison between $\mathcal{R}_2$ and $\mathcal{R}_3$ under Different $G(0.1)$ Values and SC Operations

difference between $TSC_w^{\mathcal{R}_2}$ and $TSC_w^{\mathcal{R}_3}$ is expected when shuttles are slower or lift is faster. Similarly, a smaller difference in lift waiting time between the two dispatching rules is expected if shuttles are much faster or the lift is much slower.

Even though Figure 3-2 shows that the random shuttle dispatching rule $\mathcal{R}_1$ is not the best option under classed-based storage, it is not the worst under all the scenarios. The comparisons of $\mathcal{R}_1$ against $\mathcal{R}_2$ or $\mathcal{R}_3$ under $G(0.1) = 0.2$ and 0.7 are shown in Figure 3-4. As illustrated, when $G(0.1)$ is low, $\mathcal{R}_1$ has more shuttle movement time and waiting time than $\mathcal{R}_2$. When the $G(0.1)$ value increases (i.e., the demand rates are more heterogeneous), $TSC_w^{\mathcal{R}_1} - TSC_w^{\mathcal{R}_2}$ becomes smaller, and $TSC_m^{\mathcal{R}_1} - TSC_m^{\mathcal{R}_2}$ reduced and becomes negative eventually. As Figure 3-4 shows, when $G(0.1) = 0.7$, $TSC_w^{\mathcal{R}_1} - TSC_w^{\mathcal{R}_2}$ is dominated by $TSC_m^{\mathcal{R}_1} - TSC_m^{\mathcal{R}_2}$ so that $\mathcal{R}_1$ is better than $\mathcal{R}_2$. As demonstrated by Figure 3-4, $\mathcal{R}_3$ is always superior to $\mathcal{R}_1$, and the difference between these two rules follows a very similar pattern as the difference between $\mathcal{R}_2$ and $\mathcal{R}_3$.

It is worth to note that even though the value of the differences of the expected SC travel time between the shuttle dispatching rules demonstrated by the figures are relatively low (e.g., 0.6 seconds), and it is hard to believe such a small difference can demonstrate a significant impact of the shuttle dispatching rules. However, the expected SC time only ranges from 17 seconds to 5 seconds, 0.6 seconds is a significant difference percentage-wise. As illustrated by Figure 3-2, the best shuttle dispatching rule can be at most 10.8% better than the second-best one. Moreover, a large number of tasks can be conducted in one planning horizon, and the difference will be accumulated during operations, and a significant difference (in terms of makespan reduction) can be observed. The statement is also applicable for the difference of the expected DC travel time under different combinations of policies.

3.6.2.2 Shuttle Dispatching Rule Comparison under a DC Cycle

This study also compared the two storage policies and three shuttle dispatching rules considering both retrieval and re-storage tasks executed in a DC cycle. Figure 3-5 compares the performance of the shuttle dispatching rules under different demand distributions and the number of shuttles, M, when class-based storage is used in DC operations. The best dispatching rule under each scenario is selected, and the advantage against the second-best dispatching rule is calculated through $\frac{EDC_B - EDC_{SB}}{EDC_{SB}} \times 100\%$ and listed in each cell, where EDC_B and EDC_{SB} denote the

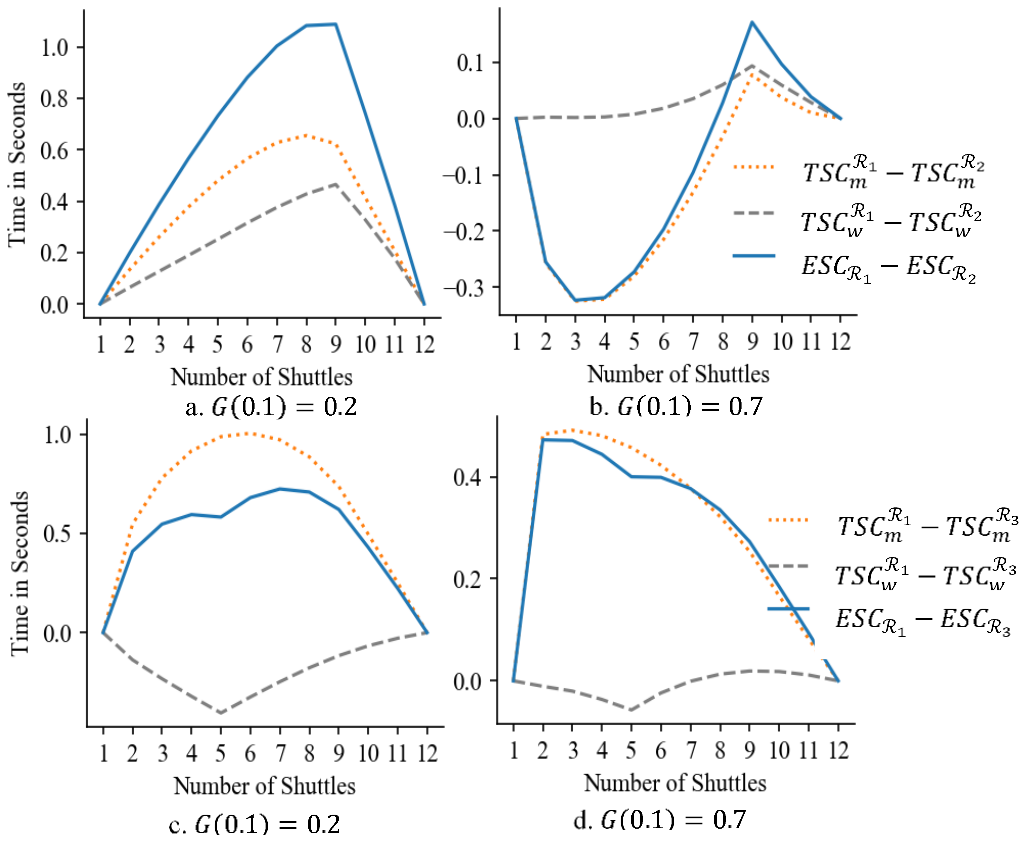


Figure 3-4. Comparison of shuttle dispatching rules under Different $G(0.1)$ Values and SC Operations

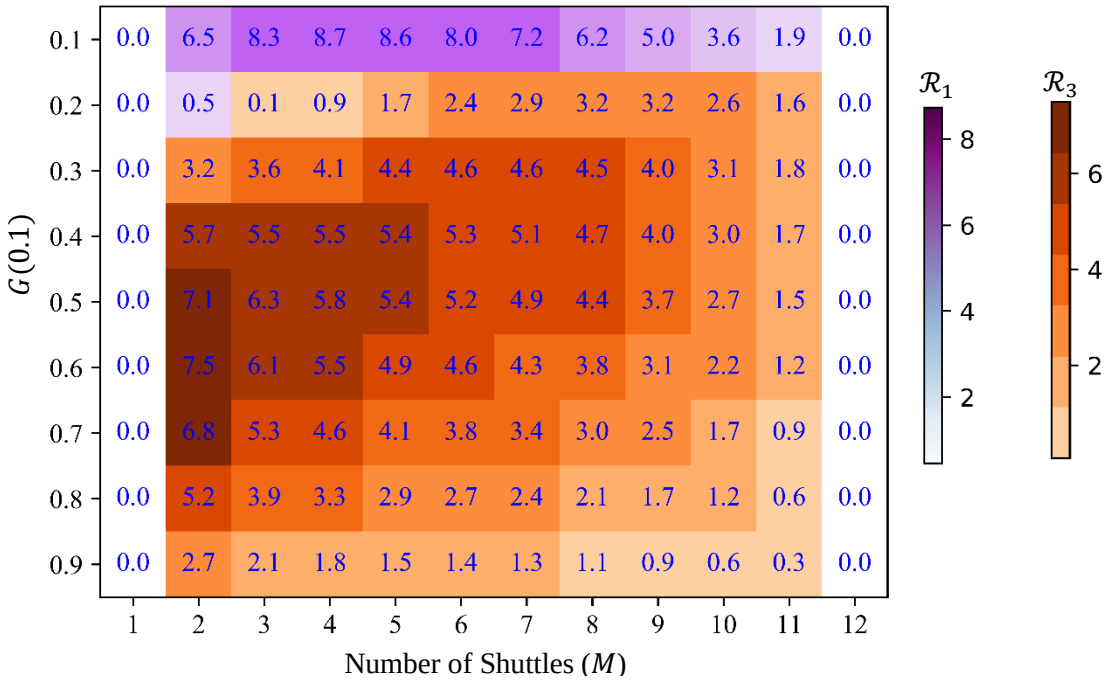


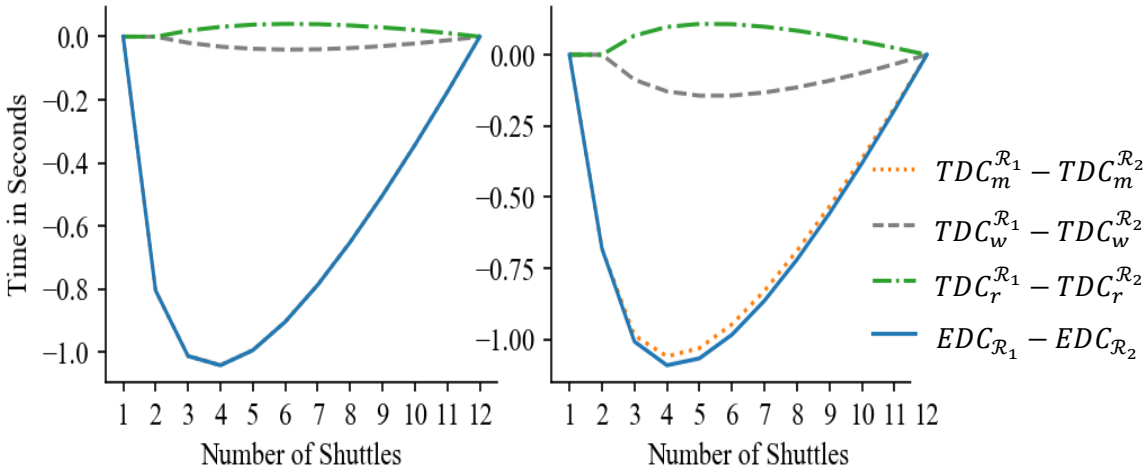
Figure 3-5. Best Dispatching Rules under Various $G(0.1)$ and M Values under Class-based Storage and DC Operations

expected DC travel time under the best shuttle dispatching rule and the second-best one, respectively. Here, purple means $\mathcal{R}_1$ is the best while red means $\mathcal{R}_3$ is the best.

- The distance-based shuttle dispatching rule $\mathcal{R}_2$ is dominated. The best dispatching rule is either $\mathcal{R}_1$ or $\mathcal{R}_3$, depending on the demand distribution and the number of shuttles M. The difference of travel times between the best and second-best dispatching rules could be as high as 8.7%. However, as discussed in subsection 3.5.2, the difference between $\mathcal{R}_1$ and $\mathcal{R}_2$ is affected by equipment's speed profile, and as discussed in Appendix G, $\mathcal{R}_1$ can be dominated by $\mathcal{R}_2$ under a different equipment speed profile.
- When the demand rate is rather homogenous (e.g., $G(0.1) = 0.1$), the random shuttle dispatching rule $\mathcal{R}_1$ is in general better than the demand-rate-based shuttle dispatching rule $\mathcal{R}_3$.
- When the demand rate is a little more heterogeneous (e.g., $G(0.1) = 0.2$), $\mathcal{R}_1$ is the best when the number of shuttles is low, while more shuttles make $\mathcal{R}_3$ more competitive.
- When the demand rate is even more heterogeneous (e.g., $G(0.1) \geq 0.3$), $\mathcal{R}_3$ is dominant, but the advantage of $\mathcal{R}_3$ over the second-best rule first increases and then decreases in M. When the demand rate is highly heterogeneous (e.g., $G(0.1) = 0.9$), the advantage of $\mathcal{R}_3$ becomes smaller because shuttles are not moved often.

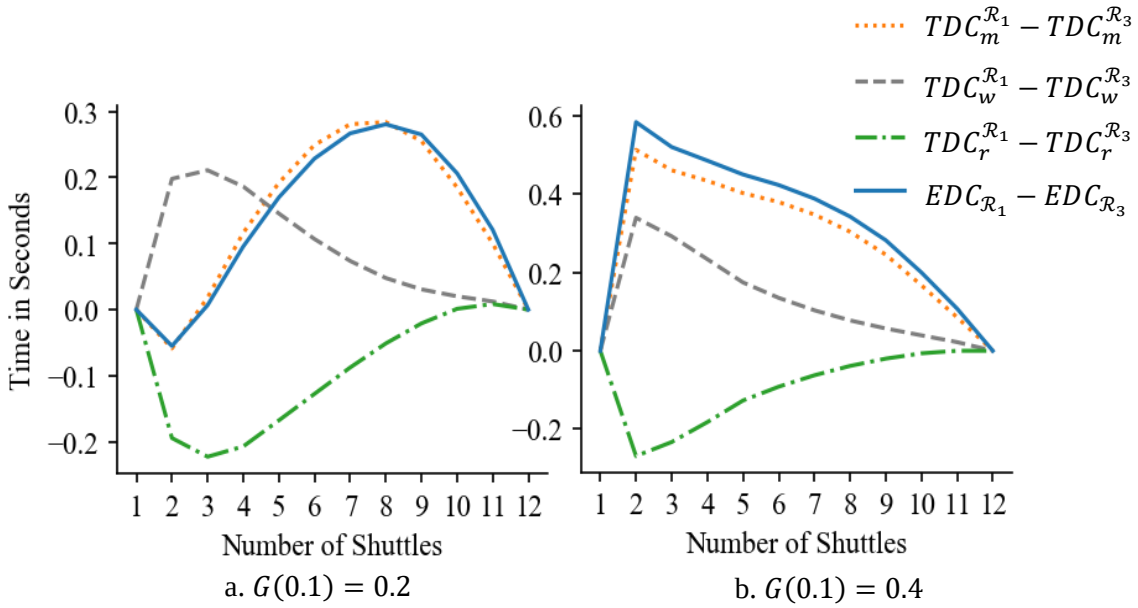
For better understanding the impacts of the shuttle dispatching rules on the expected DC travel time, the three dispatching rules under $G(0.1) = 0.2$ and 0.4 are compared in pairs in Figure 3-6, Figure 3-7, and Figure 3-8, where the expected DC travel time $EDC_{\mathcal{R}}$ is partitioned into three time components: expected time for the lift to travel between tiers for conduct re-storage and retrieval tasks ($TDC_r^{\mathcal{R}}$), expected time for moving a shuttle ($TDC_m^{\mathcal{R}}$), and expected waiting time of the lift for the shuttle to move the requested SKU ($TDC_w^{\mathcal{R}}$), where $\mathcal{R} \in \{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3\}$ represents the shuttle dispatching rule. These three terms can be represented by equation (3.26-3.28),

$$TDC_r^{\mathcal{R}} = \sum_{i=1}^{N} 2d_i t_{oi} + \sum_{i=1}^{N}\sum_{j=1}^{N} d_i d_j x'_{j|i}\, t_{ij}, \tag{3.26}$$



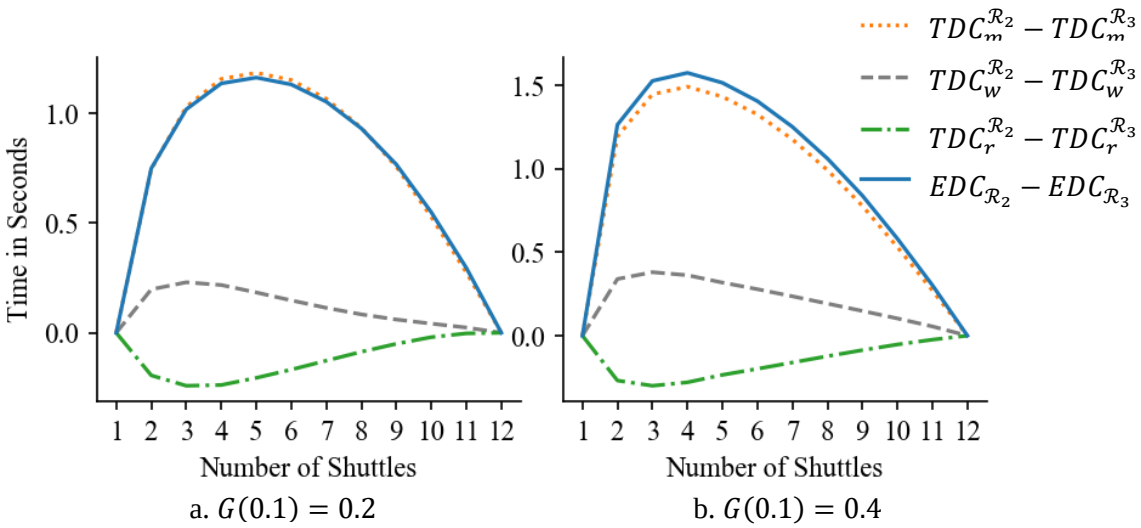
a. $G(0.1) = 0.2$ b. $G(0.1) = 0.4$

Figure 3-6. Comparison between $\mathcal{R}_1$ and $\mathcal{R}_2$ under Different $G(0.1)$ Values and DC Operations



a. $G(0.1) = 0.2$ b. $G(0.1) = 0.4$

Figure 3-7. Comparison between $\mathcal{R}_1$ and $\mathcal{R}_3$ under Different $G(0.1)$ Values and DC Operations



a. $G(0.1) = 0.2$ b. $G(0.1) = 0.4$

Figure 3-8. Comparison between $\mathcal{R}_2$ and $\mathcal{R}_3$ under Different $G(0.1)$ Values and DC Operations

$$TDC_m^{\mathcal{R}} = \sum_{i=1}^{N}\sum_{j=1}^{N} d_i d_j\,(1 - x'_{j|i})\{\sum_{\substack{k\neq j\\ k\neq i}} x_{k|(i,j)} p_{(i,j)}(k)\big(t_{ik} + t_{kj}\big) + x_{i|(i,j)} p_{(i,j)}(i)\big(t^r + t_{ij}\big)\}, \text{and} \tag{3.27}$$

$$TDC_w^{\mathcal{R}} = \sum_{i=1}^{N}\sum_{j=1}^{N} d_i d_j x'_{j|i}\left[t^r - t_{oi} - t_{ij}\right]^+ + \sum_{i=1}^{N}\sum_{j=1}^{N} d_i d_j\,(1 - x'_{j|i})t^r, \tag{3.28}$$

As illustrated in Figure 3-6, $EDC_{\mathcal{R}_1} - EDC_{\mathcal{R}_2}$ is dominated by $TDC_m^{\mathcal{R}_1} - TDC_m^{\mathcal{R}_2}$, and the difference first increases and then decreases in M. Compared with the random dispatching rule $\mathcal{R}_1$, $\mathcal{R}_2$ results in a lower expected time for moving SKUs (i.e., $TDC_r^{\mathcal{R}_1} - TDC_r^{\mathcal{R}_2} > 0$) and the difference first increases and then decreases in M. In addition, $\mathcal{R}_2$ leads to a higher expected time for the lift to wait for the shuttle to retrieve the required SKU, and $TDC_w^{\mathcal{R}_1} - TDC_w^{\mathcal{R}_2}$ first decreases and then increases in M. However, as discussed in section 3.5, $EDC_{\mathcal{R}_1} - EDC_{\mathcal{R}_2}$ is not always negative. As demonstrated in Appendix G, $EDC_{\mathcal{R}_1} - EDC_{\mathcal{R}_2} > 0$ when the lift becomes slower (i.e., $v_l = 0.5\ m/s$, and $a_l = 0.5\ m/s^2$).

Figure 3-7 compares $\mathcal{R}_1$ and $\mathcal{R}_3$. $\mathcal{R}_1$ results in a lower shuttle movement time, $TDC_m^{\mathcal{R}}$, than $\mathcal{R}_3$ when demand rate is more homogenous (i.e., $G(0.1) = 0.2$) and the number of shuttles is low (i.e., $M = 2$). $\mathcal{R}_3$ makes shuttles more likely to stay on the lower tiers, and therefore, shuttle reallocation is frequently required for tasks from higher tiers, which results in a higher $TDC_m^{\mathcal{R}}$ for $\mathcal{R}_3$ when demand rate is more homogenous (i.e, the demand for higher tiers is large). However, when the demand rate is more heterogeneous (i.e., $(0.1) = 0.4$), the disadvantage of $\mathcal{R}_3$ becomes an advantage over $\mathcal{R}_1$ (i.e., $TDC_m^{\mathcal{R}_1} - TDC_m^{\mathcal{R}_3} > 0$), since the demand are mainly from the lower tiers and the bottom $M - 1$ tiers always have a shuttle under $\mathcal{R}_3$. Moreover, $\mathcal{R}_3$ results in a lower expected waiting time for the lift to get the required SKU (i.e., $TDC_w^{\mathcal{R}}$) and a higher expected time for the lift to move SKUs (i.e., $TDC_r^{\mathcal{R}}$) due to less shuttle movement. Moreover, both $\left|TDC_m^{\mathcal{R}_1} - TDC_m^{\mathcal{R}_3}\right|$, $\left|TDC_w^{\mathcal{R}_1} - TDC_w^{\mathcal{R}_3}\right|$, and $\left|TDC_r^{\mathcal{R}_1} - TDC_r^{\mathcal{R}_3}\right|$ increase and the decrease in M. Figure 3-8 compares $\mathcal{R}_2$ and $\mathcal{R}_3$. As demonstrated, $\mathcal{R}_2$ is always worse than $\mathcal{R}_3$ (i.e., $EDC_{\mathcal{R}_2} -$

$EDC_{\mathcal{R}_3} > 0$) and the difference between these two rules follows a very similar pattern as the difference between $\mathcal{R}_1$ and $\mathcal{R}_3$.

3.6.3 Storage Policy Comparison

The comparison between random (i.e., $G(0.1) = 0.1$) and class-based storage policies under each shuttle dispatching rule under SC and DC operations are illustrated in Figure 3-9 and Figure 3-10 with $G(0.1) = 0.2$. The differences between random and class-based storage policies under other $G(0.1)$ values follow the same pattern. Even though random storage results in less waiting time under all shuttle movement dispatch rules (except under $\mathcal{R}_3$ and DC operations) and less shuttle reallocation time under $\mathcal{R}_2$ and SC operations, random storage usually yields much higher expected SKU retrieval and re-storage time, $TSC_r^{\mathcal{R}}$ or $TDC_r^{\mathcal{R}}$, which dominates the expected travel time. Therefore, random storage is always worse than class-based storage policy, which is consistent with previous studies on storage policies (Yu et al., 2015). However, it does not mean that class-based storage is always the best storage policy. A third storage policy that is better than both even under the demand-rate-based shuttle dispatching rule $\mathcal{R}_3$ might exist.

A counter example against the statement that class-based storage is always the best is given in Table 3-2. This simple system consists of three tiers and two shuttles and adopts $\mathcal{R}_3$ and SC operation with only retrieval tasks. The three demand rates are 0.34, 0.33, and 0.33. The time parameters (in seconds) are $t_{o1} = 0.577$, t_{o2}= 0.816, $t_{o1} = 1$, $t^r = 1$ and

$$t_{ij} = \begin{cases} 0, & i = j \\ 0.577, & |i-j| = 1 \\ 0.816 & |i-j| = 2. \end{cases}$$

The first row shows the performance of class-based storage, where $d_1 = 0.34$ and $d_2 = d_3 = 0.33$. The second row shows another storage situation in which the highest demand row is moved to the top, i.e., $d_1 = d_2 = 0.33$ and $d_3 = 0.34$. This movement reduces waiting time, $TSC_w^{\mathcal{R}_3}$, without hurting the time to move SKUs, $TSC_r^{\mathcal{R}_3}$, too much. Therefore, the travel time, $ESC_{\mathcal{R}_3}$, is reduced. Please note that the shuttle movement time, $TSC_m^{\mathcal{R}_3}$, keeps the same under the two storage policies. Therefore, class-based storage is not always the best storage policy even under the demand-rate-based shuttle dispatching rule $\mathcal{R}_3$, which is against our intuition and justifies a

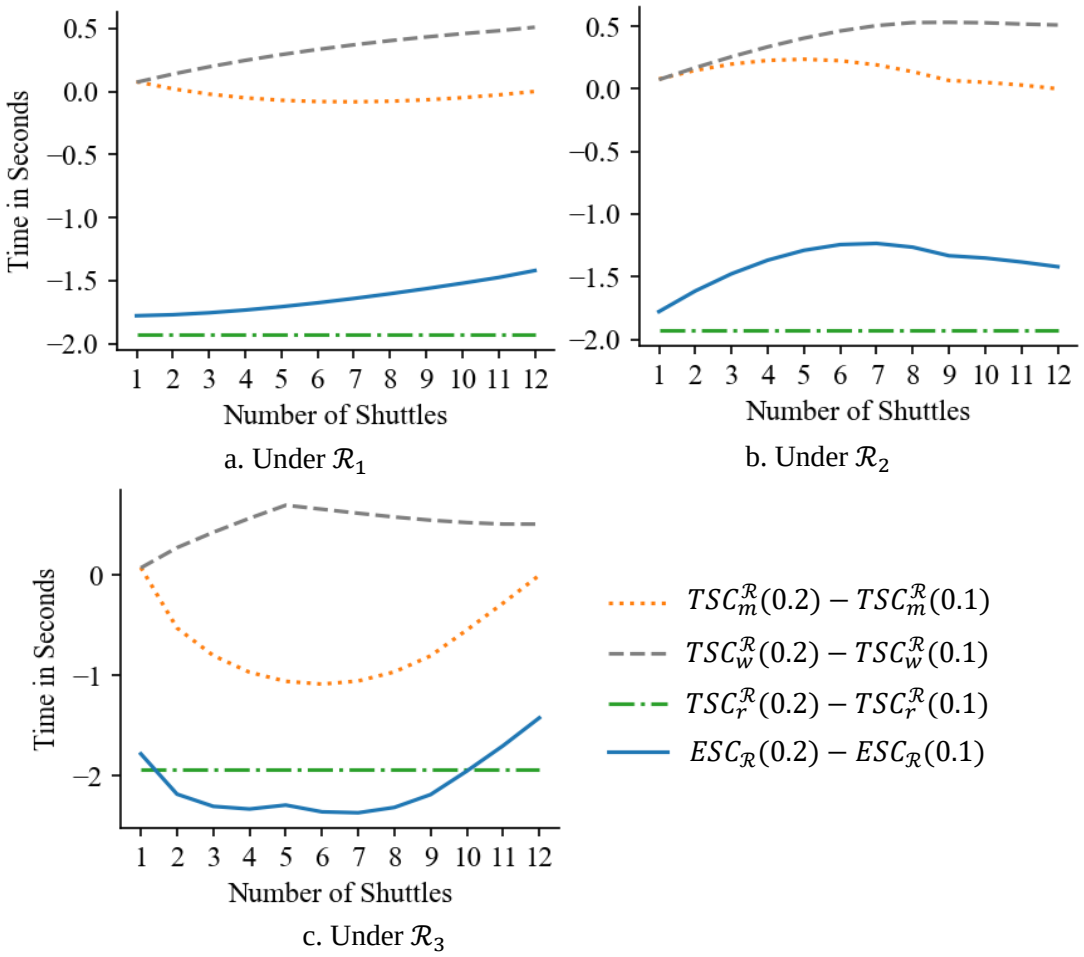


Figure 3-9. Comparison of Storage Policies under $G(0.1) = 0.2$ and SC Operations

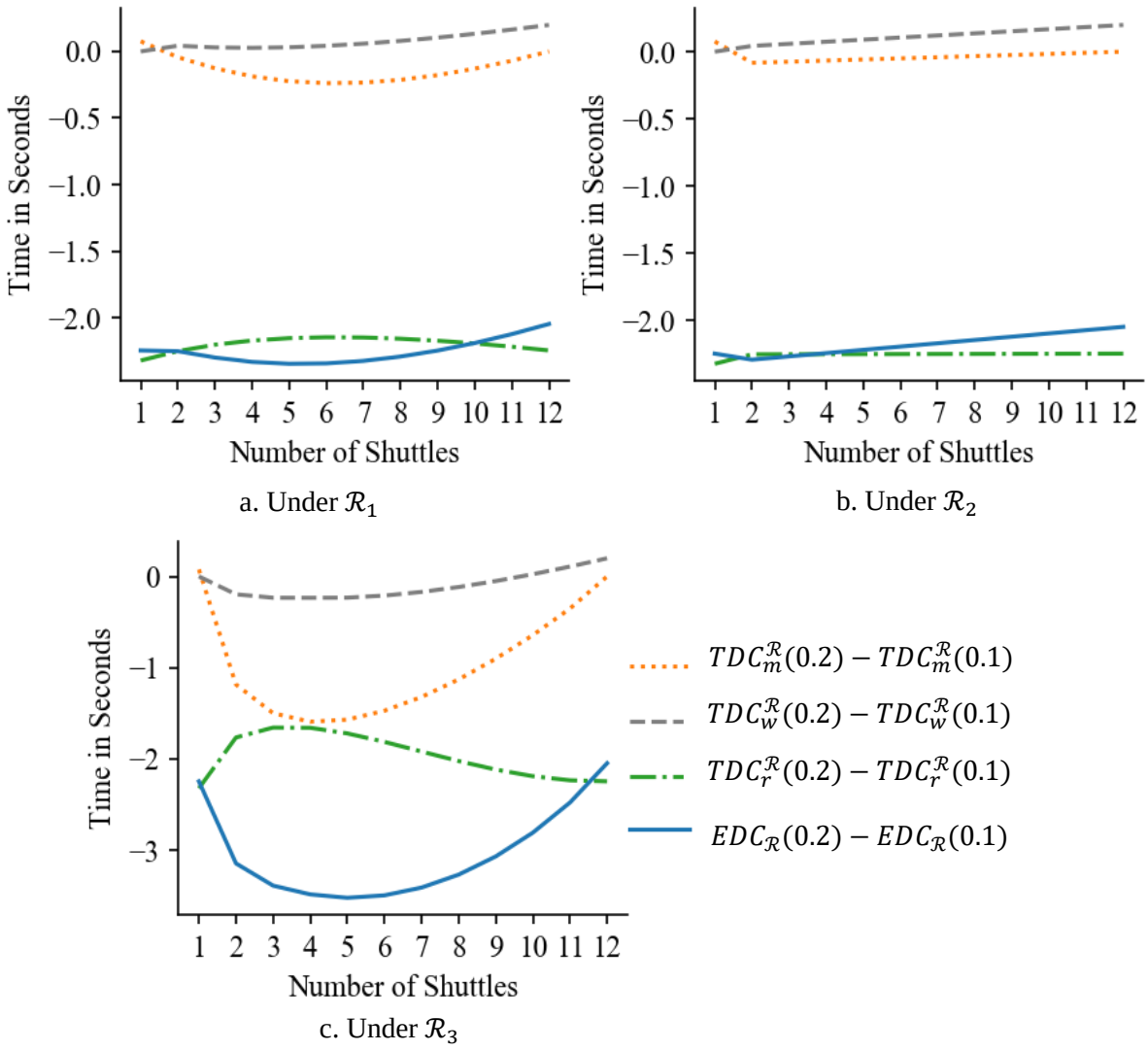


Figure 3-10. Comparison of Storage Policies under $G(0.1) = 0.2$ and DC Operations

Table 3-2. A Counter Example Against Class-based Storage under SC Operations

	$ESC_{\mathcal{R}_3}$	$TSC_r^{\mathcal{R}_3}$	$TSC_m^{\mathcal{R}_3}$	$TSC_w^{\mathcal{R}_3}$
Class-based Storage	2.28	1.59	0.19	0.50
After Storage Reallocation	2.22	1.60	0.19	0.43

need to carefully study storage policy. Please note that class-based storage definitely minimized the expected SKU retrieval time, $TSC_r^{\mathcal{R}}$, under any shuttle dispatching rules.

3.7 Conclusion

This study examines different storage assignment policies and shuttle dispatching rules in a tier-to-tier SBS/RS operating in SC or DC operations. The system allows fewer shuttles than tiers to decrease the cost and improve flexibility. This study developed travel time models under each combination of storage and dispatching rules to calculate the expected travel time under SC and DC cycle, respectively. The system is modeled as a DTMC to calculate shuttle distribution under different operational policies. Based on the shuttle distribution, expected travel time models are developed and validated through simulation. The expected travel times models developed by this study can help a warehouse designer or manager to decide the most appropriate storage assignment and shuttle dispatching rule as well as the number of shuttles in their system.

Numerical experiments are conducted and provide the following observations.

- The demand distribution information is important for selecting the appropriate storage policy and shuttle dispatching rule.
- Class-based storage is always better than random storage while the best dispatching rule is affected by demand distribution patterns, represented by the $G(0.1)$ value, the number of shuttles, and operational cycle (i.e., SC and DC).
- When class-based storage is applied but the demand rate does not vary a lot across classes, the demand rate-based shuttle dispatching rule $\mathcal{R}_3$ is the best if there are not many shuttles when the system operates in an SC cycle with only retrieval tasks considered. However, when the system operates in a DC cycle, the random shuttle dispatching rule $\mathcal{R}_1$ or distance-based shuttle dispatching $\mathcal{R}_2$, depending on the equipment speed profile, might be better than $\mathcal{R}_3$ if there are not many shuttles.
- If demands are more heterogeneous, $\mathcal{R}_3$ becomes more competitive in every scenario. However, having more shuttles beyond a small number (e.g., $M \geq 4$) will reduce the advantage of $\mathcal{R}_3$, and under SC operations, it may be defeated by the distance-based shuttle dispatching rule $\mathcal{R}_2$ when $G(0.1)$ is low (e.g., $G(0.1) < 0.4$).
- For warehouse designers or managers with limited or no demand information, which implies only random storage policy is applicable, the distance-based shuttle dispatching

rule $\mathcal{R}_2$ is always better than random dispatching rule $\mathcal{R}_1$ under SC operations. However, under DC operations, depending on the equipment's speed, $\mathcal{R}_1$ can be better than $\mathcal{R}_2$.

Several future studies can further provide insights for tier-to-tier SBS/RS and facilitate their designs and operations. Even though this study considered both SC and DC operations, we only included retrieval tasks and re-storage tasks, so that shuttle reallocation is not needed for storage tasks. However, in practice, other types of storage tasks may happen (e.g., replenishment), and it is very likely that the storage tier also needs a shuttle. Therefore, future studies may consider replenishment tasks. As discussed in section 3.4, instead of dividing rack into classes along the vertical direction, many previous studies classify storage locations into classes based on their distances to the lift/buffer stations. Even though these two classification strategies have never been compared, we assume the efficiency of these two classification strategies will be affected by both equipment speed profile, equipment numbers, and demand pattern. Therefore, a comprehensive comparison between these two classification strategies and an investigation into the optimal number and boundaries of classes under different scenarios are worth future research. Besides, similar to many other studies, this study ignored the situation when an SKU is potentially blocked by earlier SKUs from the same tier to simplify the analysis. A future study considering the blocking effect will be interesting. Furthermore, although the developed travel time models can facilitate system designs and implicitly incorporates some physical design parameters, such as rack dimension and equipment speed, in travel times, this study did not optimize those design parameters for the best performance. However, having those physical design parameters as variables is expected to bring strong non-linearity and result in great computational challenges.

CHAPTER FOUR

AUTOMATED WAREHOUSE DESIGN CONSIDERING 2D AND 3D AS/RS

4.1 Abstract

Numerous AS/RS variants have been introduced for different business needs. However, there is a lack of tools for decision-makers to select and design the best AS/RS for their business needs. This paper is the first to consider the warehouse design problem with different application scenarios (characterized by different distributions of inventory levels and demand rates of different SKU types) by considering 2D AS/RS and 3D AS/RS as options to reduce the warehouse investment while maintaining a certain level of throughput capacity. The warehouse design problem is first modeled as mixed-integer nonlinear programming and then is converted to mixed-integer programming based on optimality conditions. A branch-and-bound algorithm is developed for the computational challenges and is further modified to reduce the solving time. Numerical experiments showed the impacts of the cost parameters, and the distribution of the inventory levels and demand rates of different SKU types on warehouse design. A high cost/penalty of not meeting the desired throughput capacity will lead to a warehouse design with a large number but shallower racks. Having a high land/space and rack cost will have the same effect while having a high equipment cost will have contradictory impacts on warehouse design. When the distribution of demand rates across SKU types is homogeneous, a heterogeneous distribution of inventory levels will lead to a warehouse design where the racks' depth is more unevenly distributed. The impact of the distribution of the inventory levels will be eliminated if the distribution of demand rates across SKU types is heterogeneous or the cost of not meeting the expected throughput capacity is high. Heterogeneous distribution of demand rates or a high cost of not meeting the required throughput capacity will also lead to a warehouse design with a large number but shallower racks, and racks' depth will be more unevenly distributed. Surprisingly, the warehouse design problem can be divided into two steps: 1) solve the warehouse design problem to minimize the total depth of racks iteratively on the different numbers of racks; 2) solve the product allocation problem with all feasible solutions from step one and select the one with the minimum objective value. At last, 2D AS/RS racks are only used in the case with an extremely high cost of throughput capacity, which indicates the advantages of 3D AS/RS.

4.2 Introduction

Warehouse systems/distribution centers/ fulfillment centers have been an essential component in any supply chain, and for decades, technologies have been developed to modernize warehouse

systems (Hamzaoui et al., 2021). Introduced in the 1950s, automated storage and retrieval system (AS/RS) have been widely accepted to substitute the traditional manual (picker-to-parts) warehouse system to improve system performance and reduce labor and space cost (Azadeh et al., 2019a; Boysen et al., 2019; Dong et al., 2021; Liu et al., 2020; Roodbergen and Vis, 2009). As shown in Figure 1-1, a traditional AS/RS is composed of storage racks with storage cells to accommodate stock-keeping units (SKUs), storage/retrieval (S/R) machines that can move along the aisles between two racks to execute storage and retrieval tasks, and input/output (I/O) positions where the retrieved SKUs are dropped off and incoming SKUs are picked up by an S/R machine. Traditionally, the storage rack can be single-deep or double-deep (2D), and each SKU can be accessed by an aisle-captive crane that can move along vertical and horizontal directions simultaneously. Even though 2D AS/RS has significantly improved land utilization in warehouses, the aisles between racks can still consume about 35% of land space (Hamzaoui et al., 2021; Stadtler, 1996), which also results in a long transaction time. Furthermore, the application of aisle-captive cranes requires a high investment and may reduce system flexibility (Ha and Chae, 2019; Tappia et al., 2017).

To improve the performance of AS/RS and extend the applicability of AS/RS to numerous business scenarios, AS/RS have gone through many alternatives in terms of different rack dimensions and S/R machines (Figure 1-2), such as 3D AS/RS, shuttle-based system (SBS/RS), mobile rack system, and multi-aisle system (Azadeh et al., 2019a). Even though various AS/RS are available in practice, most of the AS/RS related literature mainly focused on the physical design and operation of a few AS/RS systems with the assumption of a single-rack system (Azadeh et al., 2019a; Boysen et al., 2019; De Koster et al., 2017; Hamzaoui et al., 2021; Roodbergen and Vis, 2009). Very limited attention has been paid to AS/RS comparison, selection, and design. The system comparison studies mainly focused on the comparison of 2D AS/RS and SBS/RS by assuming pre-defined systems' design and operation policies (Ekren and Heragu, 2012), and therefore cannot provide enough insights for AS/RS selection and design. Moreover, the system selection studies also assume the predefined physical design, operation policies, or performance of the potential AS/RS options, which is far away from the practice (Azadeh et al., 2019a; Roodbergen et al., 2015).

According to our best understanding, no study particularly focused on the warehouse design problem by selecting and designing different AS/RS choices. However, such a study is necessary since the investment of a warehouse is expensive and the physical design is usually irreversible (Azadeh et al., 2019b; Roodbergen and Vis, 2009). To bridge this research gap and facilitate the design of automated warehouse systems under different application environments, this study considers warehouse design with multiple AS/RS choices to minimize the investment (e.g., space cost, equipment cost, and operation cost) while maintaining the desired throughput capacity (measured by expected cycle time; the average time for finishing one transaction) of a warehouse characterized by the distribution of inventory levels and demand rates of different SKU types. Moreover, unlike most of the previous studies that only select one AS/RS type, multiple AS/RS types can be selected, and the SKU-to-rack/system assignment will be considered. However, this study cannot include all of the AS/RS shown in Figure 1-2, and we decided to focus on the warehouse design problem by only considering the traditional 2D AS/RS and 3D AS/RS, which are popular in practice. In the future, more candidate systems can be incorporated into the model proposed by this study.

3D AS/RS, also known as multi-deep or compact storage systems, consists of racks with deep lanes that can accommodate multiple SKUs to improve space utilization and reduce the requirement on cranes. In a 3D AS/RS, an SKU can be accessed by the cooperation of S/R machines and depth move mechanisms (DMMs). Based on the types of S/R machines and DMMs, 3D AS/RS can be classified into three categories:1): crane-based 3D AS/RS with conveyor-based DMMs, 2): crane-based 3D AS/RS with shuttle-based DMMs, and 3): lift-based 3D AS/RS with shuttle-based DMMs (also called as compact SBS/RS).

In a crane-based 3D AS/RS (Figure 2-1) with conveyor-based DMMs, aisle-captive cranes are used for vertical and horizontal movement (x- and y-direction). Each lane in the 3D rack is equipped with a pair of gravity or powered conveyors as DMMs. In the system with gravity-conveyor-based DMMs, a lifting mechanism is needed for each lane to move SKUs and the storage capacity is limited. Even though powered conveyors can eliminate the limitation on storage capacity and the requirement of lifting systems, equipping each lane with powered conveyors is usually expensive (Azadeh, et al., 2019; Tappia et al., 2017). In a crane-based 3D AS/RS with shuttle-based DMMs, shuttles are used as DMMs. Such a 3D system allows fewer shuttles than the number of lanes, and the shuttles can be reallocated across lanes with the help of

cranes. The possibility of adding/decreasing shuttles can significantly improve the flexibility and reliability of 3D systems than the system with conveyor-based DMMs. Moreover, the crane-based 3D AS/RS with conveyor-based DMMs can be considered as the 3D system with shuttle-based DMMs where each lane is equipped with a shuttle.

Instead of cranes, lifts are used as S/R machines in a lift-based 3D AS/RS (Figure 4-1). The system consists of multiple tiers of multi-deep storage lanes where SKUs are stored. Lifts are responsible for vertical movements between tiers. The cooperation of "specialized" shuttles and a transfer car may realize the horizontal movements of SKUs, where shuttles move along z-direction and the transfer car moves along x-direction. Another possible horizontal movement mechanism is the utilization of 'generic' shuttles moving in both x- and z-directions. Similar to crane-based 3D AS/RS with shuttle-based DMMs, lift-based 3D AS/RS can improve system flexibility by adding or decreasing shuttles in the system. However, if shuttles and transfer cars are adopted for horizontal movement in each tier, more movement elements (lifts, shuttles, and transfer cars) need to cooperate for storage and retrieval tasks, which increases the operational complexity. In addition, "generic" shuttles are much more expensive and require proper coordination to avoid collisions, adding complexity into operations versus crane-based 3D AS/RS.

Due to the advantages of crane-based 3D AS/RS with shuttle-based DMMs over the other 3D AS/RS in terms of system flexibility, operation complexity, and hardware cost, we narrow down the 3D AS/RS to the crane-based 3D AS/RS with shuttle-based DMMs and use 3D AS/RS and crane-based 3D AS/RS with shuttle-based DMMs interchangeably in this paper. Considering different business needs, this study aims to model the warehouse design problem with 2D AS/RS and 3D AS/RS as candidates to determine the physical layout (i.e., number of racks and layout of these two systems) and product allocation (SKU-rack/system assignment). The warehouse system design problem is first modeled as a mixed-integer nonlinear programming (MINP). Optimality conditions have been proved and based on which, the MINP is converted to mixed-integer programming (MIP). Branch-and-bound algorithms are developed for computational challenges.

The remainder of this chapter is organized into six sections. Section 4.3 reviews previous studies related to the AS/RS selection and design problems. Section 4.4 gives an overview of the operational mechanism of the candidate systems and creates a MINP to describe the warehouse

Figure 4-1. An Overview of Lift-based 3D AS/RS

design problem. In section 4.5, optimality conditions are proposed and the MINP is converted into MIP, and two Branch-and-Bound (B&B) algorithms are designed to reduce the computational complexity while guarantying the optimality. In section 4.6, the warehouse design model is validated through numerical experiments, and more numerical experiments are conducted for sensitivity analysis to provide management insights. Finally, section 4.6 concludes the study and discusses the future work.

4.3 Literature Review

4.3.1 Warehouse Design Considering Multiple AS/RS Choices

Warehouse design means the identification of physical design and operation policies to optimize one or many performance metrics for a warehouse (Hamzaoui et al., 2021). The system performance is usually evaluated by hardware cost, throughput capacity (measured the expected cycle time), space utilization, orders waiting time, energy consumption, etc. In general, a warehouse design problem includes identifying and sizing multiple function areas, selecting equipment, determining the layout and operation policies for each functioning area. However, this study only focuses on the design of the storage department, and only reviews the studies considering the comparison, selection, and design of AS/RS. We recommend Gu et al. (2010), Davarzani and Norrman (2015), Bottani et al. (2019), and Kumar et al. (2021) for comprehensive reviews of warehousing researches.

Even though the selection and design of the appropriate AS/RS affect the overall warehouse performance, existing studies are preliminary and only considered AS/RS selection by assuming the physical design, operation policies, and performance of the candidate AS/RS systems are known in advance (Azadeh et al., 2019a; Baker and Canessa, 2009; Gu et al., 2010; Roodbergen et al., 2015; Rouwenhorst et al., 2000; Sprock et al., 2017). Most studies for AS/RS selection, design, and operation follow a step-by-step procedure and are empirical- or knowledge-based (Baker and Canessa, 2009; Sprock et al., 2017). Mathematical models and algorithms for warehouse design problems considering multiple AS/RS options are very rare. Changpeng Shen et al. (2010) focused on a warehouse design problem considering manual and semi-automated warehouse systems as potential options to minimize pickers' travel time. However, they assumed all the SKUs are identical and did not have to consider the SKUs-to-system assignment. In addition, as stated by Azadeh et al. (2019), Pazour and Meller (2014) is the first one to model the

warehouse design problem considering multiple order fulfillment technologies (e.g., AS/RS and manual warehouse system). However, they assumed the physical design, operation policies, and performance (i.e., throughput capacity) of each technology are known in advance, and only considered the technology selection and SKU-to-system assignment. Roodbergen et al. (2015) considered a warehouse design problem by allowing the mixed use of aisle-captive and aisle-to-aisle 2D AS/RS via simulation, but did not consider the SKU-to-system assignment.

Except for the limited attention paid to the selection of AS/RS, a few studies have been done to compare a few AS/RS systems. For instance, Heragu et al. (2011) modeled the automated vehicle storage and retrieval system (AVS/RS) and the conventional 2D AS/RS as open queuing networks and compared the performance of these two systems under serval pre-determined physical design plans and operation policies. Later, Ekren & Heragu (2012) compared the performance of AVS/RS with the aisle-to-aisle 2D AS/RS, which allows a crane to serve different racks, in terms of different performance metrics, by simulating these two systems under different configurations. However, similar to Heragu et al. (2011), this study assumed the configurations of the two systems are known in advance. The tier-to-tier AVS/RS and tier-captive AVS/RS are also compared by Küçükyaşar et al. (2021) and the similar assumptions with the previous system comparison studies are adopted, and therefore, limited its impacts on AS/RS selection and design. Moreover, all of the system comparison studies did not consider the SKU-to-system assignment problem.

4.3.2 Design and Operation of Single AS/RS systems

Different from the warehouse design problem with the selection and design of multiple AS/RS choices, numerous studies considering the system performance analysis, design, or operation of some specific AS/RS have been done. We recommend Azadeh et al. (2019a), Boysen et al. (2019), Boysen and Stephan (2016), and Roodbergen and Vis (2009) for comprehensive reviews.

Current researches on AS/RS can be categorized into three categories: system performance analysis, system configuration (i.e., rack dimensions and equipment selection), and operation strategies; studies within both of these categories usually consider a single-rack AS/RS through analytical modeling and simulation (Azadeh et al., 2019a; Boysen et al., 2015, 2019; Boysen and Stephan, 2016; Roodbergen and Vis, 2009). System analysis articles focus on estimating systems' performance in terms of one or more performance metrics (e.g., average cycle time, energy consumption, and transaction waiting time). Cycle time models, which are usually simple and

computationally friendly, are commonly used to give the closed-form formulation for estimating the average time for finishing one transaction (i.e., a retrieval task and/or a storage task), throughput capacity, and equipment utilization rate (Hamzaoui et al., 2021; Manzini et al., 2016; Xu et al., 2015, 2018, 2019b, 2019a). Queueing models and simulation models are also used to measure the other performance metrics (e.g., transactions' waiting time) in an AS/RS (Cai et al., 2014; Lerher et al., 2015b; Roy et al., 2017; Zou et al., 2016). Design optimization problems mainly consider physical configuration-related problems such as the dimension of a storage rack (Bortolini et al., 2015; De Koster et al., 2008; Hamzaoui et al., 2021; Wu et al., 2019; Xu et al., 2019b; Yang et al., 2015a, 2017; Yu and De Koster, 2009a; Zaerpour et al., 2013) and the number and locations of I/O positions (Manzini et al., 2016; Roy et al., 2015). Operation strategies related problems can be further divided into two groups: 1) long-/mid-term operation strategies such as storage policy (i.e., in rack SKU assignment) (Guo et al., 2016; Hausman et al., 1976; Ramtin and Pazour, 2015; Roshan et al., 2019; Yu et al., 2015; Yu and De Koster, 2009b; Zaerpour et al., 2013), product allocation (SKU-to-Rack assignment, which has rarely been studied), and equipment operation mechanism (e.g., shuttle assignment rules, single- or dual-command operations) which are usually studied together with system performance analysis; and 2) short-term operation policies such as task batching and sequencing (Dong et al., 2021; Han et al., 1987; Yu and De Koster, 2012), and dwell point policy (Bozer and White, 1984; Roy et al., 2015; Sari et al., 2005). In this study, we consider the number and dimension of 2D and 3D AS/RS racks, and the product allocation across systems as decision variables (i.e., 2D and 3D racks). The other operation policies do have significant impacts on warehouse design and can be determined later after the warehouse design is obtained (Gu et al., 2010).

As abovementioned, researchers usually focus on some specific AS/RS with a single-rack assumption, and cannot be directly used for practitioners for the overall design of a warehouse which consists of multiple racks and AS/RS options. A few studies have considered the design of an AS/RS consisting of multi-racks. Bozer and White (1990) considered the warehouse design problem with 2D AS/RS to determine the dimensions and number of racks in a warehouse to minimize the hardware cost while meeting a given throughput capacity. However, they assumed random product allocation and did not consider the demand rate and inventory level of different SKU types in the warehouse. Manzini et al. (2006) considered the warehouse design problem with 2D AS/RS by determining the number of aisles to minimize the transaction time without

considering the product allocation problem (i.e., SKU-to-rack assignment). Rao and Adil (2013) also considered the warehouse design problem of multi-2D AS/RS racks. However, again, they assumed identical SKUs and did not consider the production allocation (i.e., SKU-to-racks assignment). Mital et al. (2015) studied the warehouse design problem with 2D AS/RS by selecting the design from a predefined set of configurations to minimize the cost and system risk. Similar to the previous studies, they assumed the design is known in advance and did not consider the product allocation problem.

As discussed, the AS/RS design problem allowing multi-racks has only been done considering 2D AS/RS and is usually limited to determining the number and dimensions of racks without considering the warehouse characteristics (e.g., demand rate and inventory level of different SKU types) and SKU-to-rack allocation. Moreover, even though 3D AS/RS have also received some attention, researchers mainly focused on a single-rack system with conveyor-based DMMs or lift-based 3D AS/RS. Based on our best understanding, only two studies (Dong et al., 2021; Zaerpour et al., 2015) of crane-based 3D AS/RS with shuttle-based DMMs are published but only focused on the operation strategies. In summary, literature related to warehouse design considering different AS/RS options is very limited, and the warehouse design problem considering the traditional 2D AS/RS and the increasingly popular 3D AS/RS has never been considered and will be handled in this study.

4.4 Problem Description and Formulation

4.4.1 System Operation Mechanism and Assumptions

In this study, the warehouse design problem with 2D AS/RS and 3D AS/RS as candidates under different business needs is considered. The objective is to minimize the warehouse investment (e.g., land, hardware, operations, and maintenance cost) while satisfying the requirement of the system's throughput capacity. The throughput capacity is measured by the expected cycle time. Usually, cranes in 2D AS/RS and 3D AS/RS can operate in two modes: 1) A Single-command (SC) Cycle: A crane executes one retrieval or one storage task in one operational cycle, and 2) A Dual-command (DC) Cycle: A crane executes one retrieval and one storage task in one operational cycle. In many business scenarios, especially online retailing that requires a high responsiveness level, retrieval tasks are more critical, and the retrieval and storage tasks can be performed in different time windows (Dong et al., 2021; Liu et al., 2020; Wang et al., 2015).

Therefore, this study only considers the expected cycle time for executing a retrieval task in an SC cycle. The expected cycle time models for 2D and 3D AS/RS will be developed in subsection 4.4.2. The different business needs are represented by the distribution of inventory levels and demand rates of different SKU types as well as the desired throughput capacity (i.e., desired average cycle time), and the product allocation (i.e., SKU-to-rack assignment) is also included as a decision variable. Moreover, all SKUs require the identical size of storage cells. We assume the random SKU-in-rack storage, which is a common assumption in warehouse design problems, to reduce the problem complexity (Bozer and White, 1990; Ha and Chae, 2019; Tappia et al., 2017, 2019). Moreover, according to Zaerpour et al. (2013), the in-rack storage policy does not affect the optimal rack configuration. In addition, we assume that the SKUs assigned to a 3D rack cannot share a storage lane with other types of SKUs. In another word, each lane in a 3D rack can only accommodate one type of SKUs. It seems like that the exclusive SKU-to-lane assignment will significantly increase the required storage space. However, if a storage lane is allowed to accommodate various types of SKU, the blocking effect will happen and will require additional storage lanes for reshuffles, and the expected cycle time for executing a retrieval task will be significantly increased (Xu et al., 2019b).

Moreover, instead of simply comparing the 2D and 3D AS/RS and selecting one system for the warehouse, this study allows the mixed-use of 2D AS/RS and 3D AS/RS racks. In addition, as presented in most of the warehouse design studies (Bozer and White, 1990; Lerher et al., 2015b; Liu et al., 2020), this study assumes that all racks are identical in x and y direction and are parallel placed to each other. This study also assumes that each rack, if designed, is equipped with an aisle-captive crane and an I/O point is equipped at the left lower corner of each rack. Moreover, this study allows fewer shuttles than the number of lanes in a 3D AS/RS, so that a crane also has to reallocate shuttles across lanes if necessary.

For a retrieval task in a 2D AS/RS rack, the crane departs from the I/O station, travels to the target lane/storage position, loads the SKU, and travels back to the I/O station. To accomplish a retrieval task in 3D AS/RS rack, the shuttle in the lane of the requested SKU, if there is one, will move the SKU to the end of the lane, the crane will simultaneously travel to the target lane from the I/O station, and if the shuttle is ready to hand over the SKU to the crane, the crane will pick the SKU up and transfer it to the I/O station, otherwise, the crane has to wait. If a retrieval task is located in a lane without a shuttle, the crane, starts from the I/O station, has to select a shuttle

from another lane, travels to that shuttle, picks it up, moves it to the lane where the requested SKU is located and waits until the shuttle gets the SKU ready for picking up. In addition, cranes and shuttles are assumed to remain at the position where they finished their last job before performing a new task. Since we only consider retrieval tasks, cranes will remain at I/O points, and a shuttle's dwell point is the front end of its lane. All equipment is assumed to have a constant speed and the time for equipment to unload/load an SKU is constant and is ignored.

4.4.2 Development of Cycle Time Models

In this section, we described the expected cycle time models for a 2D AS/RS and a 3D AS/RS rack j under SC operations. The expected cycle time models will be validated in section 4.6. Let n_x, n_y, and k_j (i.e., depth of the rack) represent the number of storage cells along x, y, and z directions on rack j. If $k_j = 1$, it is a 2D AS/RS rack, otherwise (i.e., $k_j \geq 2$); rack j is a 3D rack and needs shuttles. Moreover, $w, h,$ and l are defined as the width (x-direction), height (y-direction), and length (z-direction) of a storage cell. All equipment (i.e., shuttles and cranes) is assumed to have a constant speed, and V_S, V_c^x, and V_c^y are defined to represent shuttles' speed and cranes' speed along x-and y-direction, respectively. In addition, the width, height, and length (in terms of time) of rack j are expressed by T_x, T_y and T_{z_j}, where $T_x = \frac{wn_x}{V_c^x}$, $T_y = \frac{hn_y}{V_c^y}$, and $T_{Z_j} = \frac{2lk_j}{V_s}$. By assuming a continuous rack and the Tchebyshev metric of cranes' travel time to reach a storage lane (Xu et al., 2019a), the expected cycle time for a 2D AS/RS rack j (EC_j^{2D}) is developed by Bozer and White (1984), and can be expressed by Equation (4.1),

$$EC_j^{2D} = T_1\left(\frac{\beta^2}{3} + 1\right) \tag{4.1}$$

where $T_1 = \max\{T_x, T_y\}$ and $\beta = \min\left\{\frac{T_x}{T^1}, \frac{T_y}{T^1}\right\}$.

However, as discussed in section 4.3, the expected cycle time model has never been developed for 3D AS/RS with shuttle-based DMMs that allows a smaller number of shuttles than lanes, and the gap is filled by this study. In a 3D AS/RS, the distribution of shuttles across lanes is affected by both in-rack storage policies and shuttle dispatching rules, where the shuttle distribution denotes the probability that lane i has a shuttle when an order comes and the conditional probability that lane j has a shuttle given lane i does not have one. In addition, when lane j has a

shuttle and lane i needs one, the probability the shuttle located in lane j is selected for reallocation is determined by shuttle dispatching rules. Suppose there are m_j shuttles in a 3D AS/RS rack j and random in-rack storage assignment and shuttle dispatching is applied, based on Theorem 3.1, the probability that a lane i on rack j has a shuttle when an order comes is $\frac{m_j}{n_x n_y}$, the conditional probability that a lane l has a shuttle given lane i does not have one is $\frac{m_j}{n_x n_y - 1}$, and the probability that the shuttle on lane l is chosen for lane i is $\frac{1}{m_j}$.

When a required SKU is from a lane with a shuttle (with a probability $\frac{m_j}{n_x n_y}$), the SC cycle is the same as the SC cycle in 3D AS/RS with conveyor-based DMMs and the expected cycle time model has been developed by De Koster et al. (2008) as

$$T_1\left(\frac{\beta^2}{6}+\frac{1}{2}\right)+T_j\left(\frac{b_j^3}{12a_j}+\frac{a_j^2}{6}+\frac{1}{2}\right), \tag{4.2}$$

where $T_j = \max\left\{T_1, T_{z_j}\right\}$, $b_j = \min\left\{\frac{T_x}{T_j^2}, \frac{T_y}{T_j^2}, \frac{T_{z_j}}{T_j^2}\right\}$, and $a_j = \left\{\frac{T_x}{T_j^2}, \frac{T_y}{T_j^2}, \frac{T_{z_j}}{T_j^2}\right\}/\{b_j, 1\}$.

When the required SKU is located in lane i without a shuttle (with the probability $1-\frac{m_j}{n_x n_y}$), the crane has to travel from the I/O station to lane l that has a shuttle, reallocate the shuttle lane i, wait at lane i until the shuttle gets the requested SKU ready for picking up. Since we assumed constant equipment speed, the crane's expected waiting time can be calculated as $\frac{T_{zj}}{2}$. With the assumption of random in-rack storage assignment and shuttle dispatching, lane i and lane l can be considered as two randomly selected storage lanes in the rack, and the crane's travel time can be expressed as Equation (4.3) (Bozer and White, 1984),

$$T_1\left(\frac{4}{3}+\frac{\beta^2}{2}-\frac{\beta^3}{30}\right), \tag{4.3}$$

Therefore, the expected cycle time for a 3D rack j is

$$EC_j^{3D} = \frac{m_j}{n_x n_y}\left[T_1\left(\frac{\beta^2}{6}+\frac{1}{2}\right)+T_j\left(\frac{b_j^3}{12a_j}+\frac{a_j^2}{6}+\frac{1}{2}\right)\right]+\left(1-\frac{m_j}{n_x n_y}\right)\left[T_1\left(\frac{4}{3}+\frac{\beta^2}{2}-\frac{\beta^3}{30}\right)+\frac{T_{zj}}{2}\right]. \tag{4.4}$$

4.4.3 Development of Warehouse Design Model

The objective of the study is to minimize the system's cost while maintaining a certain level of throughput capacity measured by the expected SC cycle time via determining the number and dimensions of racks (i.e., length, width, and depth of each rack), the number of cranes (equals to the number of racks) and shuttles, and the product allocation (i.e., SKU-to-rack assignment).

The daily cost (i.e., fixed cost, operation and maintenance cost, and the land cost for aisle) of having one shuttle and crane are denoted by C_s and C_v, respectively. We also define C_{rw} as the daily warehouse cost (land cost, rack cost, and warehouse operation cost such as lighting) per cubic meters. Moreover, N_a is defined as the maximum number of racks allowed in the system. Binary variables r_j, where $1 \leq j \leq N_a$, are defined to demonstrate if rack j is designed to accommodate SKUs (i.e., $r_j = 1$) or not (i.e., $r_j = 0$), and without loss of generality, we assume $r_j \geq r_{j+1}$. Then the daily warehouse investment can be expressed as

$$\sum_{j=1}^{N_a}(C_s m_j + C_v r_j + C_{rw} whln_x n_y k_j),$$

where m_j, n_x, n_y, and k_j are defined in subsection 4.4.2 and are integer variables in the warehouse design problem. Without loss of generality, we assume $k_j \geq k_{j+1}$. $w, h,$ and l have also been defined in subsection 4.4.2, and are considered as parameters in the warehouse design problem.

Let I represent the set of SKU types that need to be accommodated with i denoting indices. The inventory level and the normalized demand rate of SKU type i are known in advance and are represented by Q_i and d_i, where $\sum_{i \in I} d_i = 1$. We estimated a benchmark expected cycle time to meet a required daily throughput capacity and defined C_t as the daily benefit loss if the expected cycle time given by the warehouse design exceeds the benchmark expected cycle time (i.e., $benchmark$) by one second. The method for estimating C_t based on the desired throughput capacity can be found in Appendix H. Integer variables y_{ij} are used to represent the number of SKU type i assigned to rack j. Moreover, binary variables s_j are defined to show if rack j is a 2D AS/RS rack (i.e., $s_j = 1$) or not (i.e., $s_j = 0$). Then the cost for having a higher expected cycle time than $benchmark$ is expressed by

$$\sum_{i\in I}\sum_{j=1}^{N_a} d_i \frac{y_{ij}}{Q_i} C_t(EC_j - benchmark),$$

where

$$EC_j = s_j T_1\left(\frac{\beta^2}{3}+1\right) + (1-s_j)\left(\frac{m_j}{n_x n_y}\left[T_1\left(\frac{\beta^2}{6}+\frac{1}{2}\right)+T_j\left(\frac{b_j^3}{12a_j}+\frac{a_j^2}{6}+\frac{1}{2}\right)\right] + \left(1-\frac{m_j}{n_x n_y}\right)\left[T_1\left(\frac{4}{3}+\frac{\beta^2}{2}-\frac{\beta^3}{30}\right)+\frac{T_{zj}}{2}\right]\right),$$

and $\sum_{i\in I}\sum_{j=1}^{N_a} d_i \frac{y_{ij}}{Q_i} EC_j$ is the expected cycle time for retrieving an SKU from the warehouse. Since $benchmark$ is constant, it is dropped in the warehouse design model.

Integer variables n_{ij} are defined to demonstrate the number of lanes on rack j used to accommodate SKU type i. At last, M_1, M_2, and ε are two big constant numbers and a small constant number to facilitate the constraints development. The warehouse design problem is then expressed as a mixed-integer nonlinear programming (MINP) as follows,

$$min \sum_{j=1}^{N_a}(C_s m_j + C_v r_j + C_{rw} whl n_x n_y k_j) + \sum_{i\in I}\sum_{j=1}^{N_a} d_i \frac{y_{ij}}{Q_i} C_t EC_j \quad (4.5)$$

s.t. $$\sum_{j=1}^{N_a} y_{ij} = Q_i, \qquad i \in I, \quad (4.6)$$

$$n_x n_y \geq \sum_{i\in I} n_{ij}, \qquad 1 \leq j \leq N_a, \quad (4.7)$$

$$n_{ij} k_j \geq y_{ij}, \qquad i \in I, 1 \leq j \leq N_a, \quad (4.8)$$

$$\varepsilon(k_j - 1) \leq m_j \leq n_x n_y, \qquad 1 \leq j \leq N_a, \quad (4.9)$$

$$\sum_{i\in I} y_{ij} \leq M_1 r_j, \qquad 1 \leq j \leq N_a, \quad (4.10)$$

$$M_2(1-s_j) \geq k_j - 1, \qquad 1 \leq j \leq N_a, \tag{4.11}$$

$$EC_j = s_j T_1\left(\frac{\beta^2}{3}+1\right) + (1-s_j)\left(\frac{m_j}{n_x n_y}\left[T_1\left(\frac{\beta^2}{6}+\frac{1}{2}\right) + T_j\left(\frac{b_j^3}{12a_j}+\frac{a_j^2}{6}+\frac{1}{2}\right)\right] + \left(1-\frac{m_j}{n_x n_y}\right)\left[T_1\left(\frac{4}{3}+\frac{\beta^2}{2}-\frac{\beta^3}{30}\right)+\frac{T_{zj}}{2}\right]\right), \quad 1 \leq j \leq N_a, \tag{4.12}$$

$$n_x, n_y, y_{ij}, m_j, n_{ij} \in Z^+; T_1, T_j, T_x, T_y, EC_j \in R^+; 0 < \beta, a_j, b_j \leq 1;\ r_j, s_j \in \{0,1\}; 0 \leq k_j \leq \max_{i \in I} Q_i \tag{4.13}$$

The objective function (4.5) minimizes the warehouse investment and the penalty for not meeting the benchmark expected cycle time (i.e., throughput capacity). Constraint set (4.6) represents the product allocation (SKU-to-rack assignment) and makes sure all SKUs are accommodated. Constraint sets (4.7) and (4.8) guarantee enough storage spaces on each rack and only the same type of SKUs can be stored in a lane. Constraint set (4.9) assures that the number of shuttles on each rack cannot exceed the number of lanes if the rack is 3D, and if the rack is 2D, this rack should not have any shuttles. In another word, $m_j = 0$ if $k_j \leq 1$, and if $k_j \geq 2$, $1 \leq m_j \leq n_x n_y$. Constraint set (4.10) checks if a rack is designed for accommodating SKUs. Constraint set (4.11) ensures $s_j = 1$ if rack j is a 2D AS/RS rack (i.e., single-deep). Constraint set (4.12) calculates the expected cycle time of each rack. Constraint set (4.13) defines the range of each variable.

The existence of the integer variables and nonlinearity in the MINP lead to considerable computational challenges. In the following section 4.5, the MINP is converted to mixed-integer programming (MIP) based on optimality conditions, and branch and bound (B&B) algorithms are developed for the computational burdens.

4.5 Optimality Conditions, MIP Formulation, and B&B Algorithm

4.5.1 Optimality Conditions and MIP Formulation

Given the fact that, in practice, the solution space of n_x and n_y is restricted by the application environment (e.g., space limitation), the warehouse design problem can be simplified by solving the problem iteratively on a given set of potential values of n_x and n_y. Moreover, Theorem 4.1

and Theorem 4.2 are proved as optimality conditions to eliminate the nonlinearity and reduce the number of variables in the proposed MINP.

Theorem 4.1: Optimality Condition 1. *For a rack in crane-based AS/RS system, no matter what's the depth of storage lanes and the number of shuttles, under random in-rack storage assignment, the picking face should always be square in time (SIT; i.e., $T_x = T_y$ and $\beta = 1$) to minimize the expected cycle time of this rack.*

Proof. The optimality condition has already been proved for 2D AS/RS (Roodbergen and Vis, 2009), and this study only has to prove the case for 3D AS/RS.

Let $n = n_x n_y, V = \frac{whn}{V_c^x V_c^y} = \beta T_1^2$, and $T_1 = \sqrt{\frac{V}{\beta}}$, three scenarios can be considered.

Scenario 1: $T_{zj} \geq T_1, T_j = T_{zj}, b_j = \frac{\beta T_1}{T_{zj}}$ and $a_j = \frac{T_1}{T_{zj}}$.

Then EC_j^{3D} represented by Equation (4.2) can be written as Equation (4.14),

$$EC_j^{3D} = \frac{m_j}{n}\left(\sqrt{\frac{V}{\beta}}\left(\frac{1}{2}+\frac{\beta^2}{6}\right)+T_j\left(\frac{1}{2}+\frac{V\beta^2}{12T_{zj}^2}+\frac{V}{6T_{zj}^2\beta}\right)\right)+\left(1-\frac{m_j}{n}\right)\left(\sqrt{\frac{V}{\beta}}\left(\frac{4}{3}+\frac{\beta^2}{2}-\frac{\beta^3}{30}\right)+\frac{T_{zj}}{2}\right), \quad (4.14)$$

By taking the derivation of EC_j^{3D} on β, it is clear that under scenario 1, EC_j^{3D} linearly decreases in β, and therefore, to minimize EC_j^{3D}, we should have $\beta = 1$.

Scenario 2. $T_{zj} \leq T_1, T_j = T_1, b_j = \frac{T_{zj}}{T_1}$ and $a_j = \beta$.

Under this scenario, EC_j^{3D} can be formulated as (4.15), and by taking the derivation on β, it can be proved that, under scenario 2, β should also be 1 to minimize EC_j^{3D}.

$$EC_j^{3D} = \frac{m_j}{n}\left(\sqrt{\frac{V}{\beta}}\left(\frac{1}{2}+\frac{\beta^2}{6}\right)+\sqrt{\frac{V}{\beta}}\left(\frac{1}{2}+\frac{T_{zj}^3\beta^{\frac{3}{2}}}{12\beta v^{\frac{3}{2}}}+\frac{\beta^2}{6}\right)\right)+\left(1-\frac{m_j}{n}\right)\left(\sqrt{\frac{V}{\beta}}\left(\frac{4}{3}+\frac{\beta^2}{2}-\frac{\beta^3}{30}\right)+\frac{T_{zj}}{2}\right), \quad (4.15)$$

Scenario 3. T_{zj} is the second-largest among $\{T_x, T_y, T_{zj}\}$.

Let's assume $T_x \geq T_{zj} \geq T_y$, then $\beta = b_j = \frac{T_y}{T_x}, a = \frac{T_{zj}}{T_x}$, and Equation (4.2) can be rewritten to Equation (4.16).

$$EC_j^{3D} = \frac{m_j}{n}\left(T_x\left(\frac{1}{2}+\frac{T_y^2}{6T_x^2}\right)+T_x\left(\frac{1}{2}+\frac{T_xT_y^3}{12T_{zj}T_x^3}+\frac{T_{zj}^2}{6T_x^2}\right)\right)+\left(1-\frac{m_j}{n}\right)\left(T_x\left(\frac{4}{3}+\frac{T_y^2}{2T_x^2}-\frac{T_y^3}{30T_x^3}\right)+\frac{T_{zj}}{2}\right), \quad (4.16)$$

By taking the derivation of EC_j^{3D} over T_x, it can be proved that if $T_x \geq T_{zj} \geq T_y$, to minimize the expected cycle time, T_x should equal to T_{zj}. Therefore, scenario 3 is converted to scenario 1 and we should have $\beta = 1$. The same logic can be applied for the case in which $T_y \geq T_{zj} \geq T_x$. ■

Theorem 4.1 indicates that, in an optimal rack design, T_x should always equal to T_y (i.e., $\frac{wn_x}{V_c^x} = \frac{hn_y}{V_c^y}$). Therefore, variable β fixed as one, n_x and n_y can be replaced by a new integer variable n, which denotes the number of storage lanes on a rack, and the nonlinearity brought by n_xn_y can be eliminated. Please note that Theorem 4.1 is not only applicable under random in-rack storage assignment but also under the other in-rack storage policies since the optimal rack dimension is not affected by storage policies (Zaerpour et al., 2013). Moreover, in this paper, we only solved the problem with $n = n_xn_y$ and assuming $T_x = T_y$. However, it not necessary to have $T_x = T_y$. As long as the potential values of n_x and n_y can be decided, which is common in practice, variable β and $n = n_xn_y$ can be fixed.

Theorem 4.2: Optimality Condition 2. *Let n denote the number of lanes on a 3D rack j, then to minimize the shuttles cost and the cost of not meeting the desired throughput capacity of this rack, the number of shuttles on rack j can only be one or n.*

Proof. Let C_j^{st} represent the combination of the shuttles cost and the penalty of not meeting the benchmark expected cycle time of rack j, we have

$$C_j^{st} = C_s m_j + \sum_{i\in I}\frac{d_i y_{ij}}{Q_i}\left(\frac{m_j}{n}EC_{j,k_j}^{3D1}+\left(1-\frac{m_j}{n}\right)EC_{j,k_j}^{3D2}\right), \quad (4.17)$$

Where,$EC_{j,k_j}^{3D1} = T_1\left(\frac{1}{2}+\frac{\beta^2}{6}\right)+T_j\left(\frac{1}{2}+\frac{b_j^3}{12a_j}+\frac{a_j^2}{6}\right)$ and $EC_{j,k_j}^{3D2} = T_1\left(\frac{4}{3}+\frac{\beta^2}{2}-\frac{\beta^3}{30}\right)+\frac{T_{zj}}{2}$ denote the expected SC cycle time for the cases that a required SKU is in a lane with a shuttle and the required SKU is from a lane without a shuttle, and k_j represents the depth of rack j. By taking the derivation of C_j^{st} over m_j, we can have equation (4.18).

$$\frac{dC_j^{st}}{dm_j} = C_s + \frac{1}{n}\sum_{i\in I}\frac{d_i y_{ij}}{Q_i}\left(EC_{j,k_j}^{3D1} - EC_{j,k_j}^{3D2}\right), \tag{4.18}$$

It is straightforward that $EC_{j,k_j}^{3D1} - EC_{j,k_j}^{3D2} \leq 0$. Therefore, once the depth of a 3D rack j, k_j, is determined, we can either have $\frac{dC_j^{st}}{dm_j} \geq 0$ or $\frac{dC_j^{st}}{dm_j} \leq 0$, which means a 3D rack can only have one or n shuttles. ■

For any given n, EC_j in Equation (4.12) can be written as $EC_{k_j}^1 + EC_{k_j}^2$, and $EC_{k_j}^1$ and $EC_{k_j}^2$ based on different k_j can be calculated in advance as follows,

$$EC_k^1 = \begin{cases} 0 & k = 0, \\ T_1\left(\frac{\beta^2}{3}+1\right) = \frac{4}{3}\sqrt{\frac{whn}{V_c^x V_c^y}} & k = 1, \\ \frac{4}{3}\sqrt{\frac{whn}{V_c^x V_c^y}} + \frac{(2kl)^3}{V_s^3\frac{whn}{V_c^x V_c^y}} & k \geq 2, \sqrt{\frac{whn}{V_c^x V_c^y}} \geq \frac{2kl}{V_s}, \\ \frac{2}{3}\sqrt{\frac{whn}{V_c^x V_c^y}} + \frac{2kl}{V_s}\left(\frac{1}{2} + \frac{whnV_s^2}{4(2kl)^2 V_c^x V_c^y}\right) & k \geq 2, \sqrt{\frac{whn}{V_c^x V_c^y}} < \frac{2kl}{V_s}, \end{cases} \text{, and}$$

$$EC_k^2 = \begin{cases} 0 & k = 0, \\ T_1\left(\frac{\beta^2}{3}+1\right) = \frac{4}{3}\sqrt{\frac{whn}{V_c^x V_c^y}} & k = 1, \\ \frac{9}{5}\sqrt{\frac{whn}{V_c^x V_c^y}} + \frac{kl}{V_s} & k \geq 2, \end{cases}.$$

Let's introduce binary variables n_{jk}^z, where $1 \leq j \leq N_a, 0 \leq k \leq \max_{i\in I} Q_i$, and if the depth of rack j is k storage cells, $n_{jk}^z = 1$; otherwise, $n_{jk}^z = 0$. Integer variables r_j, s_j, and k_j shown in the MINP can be eliminated. In addition, the decision variable indicating the combination of the shuttles cost and the cost of not meeting the benchmark expected cycle time (or desired throughput capacity) of rack j, C_j^{st}, is defined. Continuous variables d_j, e_j, and binary variables f_j are also introduced for incorporating Theorem 4.2. Considering n as a parameter, the MINP represented by Equation (4.5-4.13) can be converted to the MIP defined by Equation (4.6) and (4.19-4.30).

$$\min \sum_{j=1}^{N_a} C_v \sum_{k=1}^{K} n_{ik}^z + C_{rw} whln \sum_{k=1}^{K} k n_{ik}^z + C_j^{st} \tag{4.19}$$

s.t. (4.6)

$$\sum_{k=0}^{\max_{i\in I} Q_i} n_{jk}^z = 1, \qquad 1 \le j \le N_a, \tag{4.20}$$

$$d_j \ge C_s + C_t \sum_{i\in I} \frac{d_i y_{ij}}{Q_i}\left(\frac{1}{n} EC_k^1 + \left(1-\frac{1}{n}\right) EC_k^2\right) - M_1(1-n_{jk}^z), \quad 2 \le k, 1 \le j \le N_a, \tag{4.21}$$

$$e_j \ge C_s n + C_t \sum_{i\in I} \frac{d_i y_{ij}}{Q_i} EC_k^1 - M_1(1-n_{jk}^z), \qquad 2 \le k, 1 \le j \le N_a, \tag{4.22}$$

$$d_j - e_j \le M_2 f_j, \qquad 1 \le j \le N_a, \tag{4.23}$$

$$e_j - d_j \le M_2(1-f_j), \qquad 1 \le j \le N_a, \tag{4.24}$$

$$C_j^{st} \ge e_j - M_2\left(1-f_j+n_{j1}^z\right), \qquad 1 \le j \le N_a, \tag{4.25}$$

$$C_j^{st} \ge d_j - M_2(f_j+n_{j1}^z), \qquad 1 \le j \le N_a, \tag{4.26}$$

$$C_j^{st} \ge \sum_{i\in I} \frac{d_i y_{ij}}{Q_i} EC_1^1, \qquad 1 \le j \le N_a, \tag{4.27}$$

$$n \ge \sum_{i\in I} n_{ij}, \qquad 1 \le j \le N_a, \tag{4.28}$$

$$kn_{ij} - y_{ij} \ge M_2\left(1-n_{jk}^z\right), \qquad i \in I, 1 \le j \le N_a, \quad 0 \le k \le \max_{i\in I} Q_i, \tag{4.29}$$

$$y_{ij}, n_{ij} \in Z^+, d_j, e_j, f_j \in R^+, n_{jk}^z \in \{0,1\}. \tag{4.30}$$

The objective function (4.19) is the same as the objective function (4.5) but grouped the shuttles cost and the cost for not meeting the benchmark expected cycle time. Constraint set (4.20) ensures the depth of a rack j can only take one value from $\{0,1,\dots,\max_{i\in I} Q_i\}$. Constraint sets (4.21-4.27) calculate C_j^{st} for each rack j, where constraint sets (4.21-4.26) indicate that if a rack is 3D, it can only have one or n shuttles, and constraint set (4.27) provides a lower bound of C_j^{st}. Constraint set (4.28) and (4.29) together ensure that only one type of SKU can be assigned to a storage lane and the assignment does not exceed the lane's storage capacity.

Even though the nonlinearity and multi-variables in the MINP have been eliminated by the MIP defined by Equation (4.6) and (4.19-4.30), the warehouse design problem, as shown in Theorem 4.3, is NP-hard. The computational challenge, as will be shown in section 4.6, are still relatively large.

Theorem 4.3. *The warehouse design problem defined by the MIP is NP-hard.*

Proof. Assuming the number and dimensions of racks and the number of shuttles on each rack are known in advance, then the problem becomes a product allocation problem. However, since the expected cycle time and the storage capacity of each rack changes in terms of product allocation, the problem can be considered as a bin-packing problem with varied cost and bin size. Since the traditional bin-packing problem is known as NP-hard (Korte and Vygen, 2012), the warehouse design problem, which is much harder than the traditional bin-packing problem, is also NP-hard. ■

4.5.2 Branch and Bound Algorithm

In this section, the concept of the branch-and-bound (B&B) algorithm is adopted for overcoming the computational challenges of the warehouse design problem. The optimal solution of the warehouse design problem can be solved by applying the proposed B&B algorithm iteratively on the potential values of n (i.e., number of lanes on each rack).

For a warehouse design problem with given n, a branch-and-bound tree can be created where each node in the tree is a list of integer numbers, and each element in the list corresponds to a rack and the value of an element represents the depth of a rack. For instance, a node $[k_1, k_2, .., k_{n_r}]$, where $1 \le k_1 \le \cdots \le k_{n_r} \le \max_{i \in I} Q_i$, and $1 \le n_r = |[k_1, k_2, .., k_{n_r}]| \le N_a$, represents a warehouse system design with n_r racks, and $k_1, k_2, .., k_{n_r}$ are the depth of each rack. Moreover, the value of n_r is the same as the layer that node is located in the branch-and-bound tree. For instance, at the first layer, $n_r = 1$, which means only one rack is designed to accommodate SKUs. Branching on this node is done by adding one more rack with depth ranges from k_{n_r} to $\max_{i \in I} Q_i$. Therefore, the branches on the node $[k_1, k_2, .., k_{n_r}]$ will be $[k_1, k_2, .., k_{n_r}] + [k_{n_r+1}]$ where $k_{n_r+1} \in \{k_{n_r}, k_{n_r} + 1, \dots, \max_{i \in I} Q_i\}$. An example to demonstrate the branch-and-bound tree is shown in Appendix I. The B&B algorithm with depth-first-search (DFS) procedure can be described step-by-step as below.

Step 1 (initialization). Solve the MIP by Gurobi, if the model is infeasible, stop. Else, set the best known upper bound (i.e., UB) and best known lower bound (i.e., LB) of the warehouse design problem as the objective value of the first feasible solution obtained by Gurobi and zero, respectively. Initialize a queue by creating a list $q = [[1], [2], \dots, \left[\max_{i \in I} Q_i\right]]$ (i.e., nodes in the first layer of the branch-and-bound tree).

Step 2. Loop until q is empty or $LB = UB$:

1. Take a node N off the queue following last-in-first-out.
2. If the lower bound obtained by node N (i.e., lb) is larger than LB, set $LB = lb$.
3. If $lb < UB$ and the node N cannot provide an upper bound (i.e., ub), branch on node N and add the newly generated nodes into q.
4. Else if $lb < UB$, and node N can provide an upper bound and $ub < UB$, set $UB = ub$ and discard node N.
5. Else if $lb > UB$, discard node N.

In the B&B algorithm, each node in q is explored to update the best known upper bound (UB) and the best known lower bound (LB) of the warehouse design problem and cut nodes and branches to reduce the computational burden. For each node N, the lower bound is obtained by solving the product allocation problem with the given configuration of racks while allowing the storage lanes to accommodate different types of SKUs and ignoring the blocking effect. For any node, $\left[k_1, k_2, .., k_{n_r}\right]$, the lower bound can be calculated following the algorithm shown below.

Step 1. Sequence all SKU types in descending order of $\frac{d_i}{Q_i}$. Set $y_{ij} = 0$ for $i \in I, 1 \le j \le n_r$ which represents the SKU-to-rack assignment (i.e., product allocation). Set the available storage space on rack j, as $S_j = nk_j$, where $1 \le j \le n_r$.

Step 2. If $\sum_{i \in I} Q_i = 0$, stop and output $\sum_{j=1}^{n_r} C_v + C_{rw} whlnk_j + C_j^{st}$, where $C_j^{st} = \min (C_s n + C_t \sum_{i \in I} d_i \frac{y_{ij}}{Q_i} EC_{k_j}^1, C_s + C_t \sum_{i \in I} d_i \frac{y_{ij}}{Q_i} (\frac{1}{n} EC_{k_j}^1 + (1 - \frac{1}{n}) EC_{k_j}^2)$ if $k_j \ge 2$, otherwise; $C_j^{st} = C_t \sum_{i \in I} d_i \frac{y_{ij}}{Q_i} EC_{k_j}^1$. If $\sum_{i \in I} Q_i > 0$ and $\max_{1 \le j \le n_r} S_j > 0$, go to step 3. If $\sum_{i \in I} Q_i > 0$ and $\max_{1 \le j \le n_r} S_j = 0$, go to step 4.

Step 3. Set $i^* = \underset{i \in I, Q_i > 0}{argmax} \frac{d_i}{Q_i}, j^* = \underset{1 \le j \le n_r, S_j > 0}{argmin} S_j$. Set $y_{i^*j^*} = \min(Q_{i^*}, S_{j^*})$, $Q_{i^*} = Q_{i^*} - y_{i^*j^*}$, and $S_{j^*} = S_{j^*} - y_{i^*j^*}$; go to step 2.

Step 4. Add an artificial rack, $n_r + 1$, with depth k_{n_r}. Set $y_{in_r+1} = Q_i$ for all $Q_i > 0$. Stop and output $\sum_{j=1}^{n_r+1} C_v + C_{rw} whlnk_j + C_j^{st}$.

The upper bound of the node N is obtained by solving the product allocation problem with the given configuration of racks and the restriction that a storage lane can only accommodate the same type of SKUs. The product allocation problem can be represented by the MIP shown in Appendix J. In this study, we solve the MIP with Gurobi, and if the problem is feasible, the optimal value is obtained as ub, otherwise; node N cannot give a ub.

In this paper, we named the B&B algorithm shown above as the standard B&B algorithm, and only the information of lower bound and upper bound are used for pruning rules, which has a limited capacity for cutting nodes/branches. Fortunately, as will be demonstrated by the numerical experiments, two properties of the warehouse design problems are observed and can be adopted as pruning rules.

Property 1. If $[k_1, k_2, .., k_{n_r}]$ *provides a feasible racks configuration of the warehouse design problem, then for any* $[k'_1, k'_2, .., k'_{n_r}, k'_{n_r+1}]$ *which also provide a feasible racks configuration, only the solution with* $\sum_{1 \le j \le n_r+1} k'_j \le \sum_{1 \le j \le n_r} k_j$ *have the opportunity to provide a lower objective value than* $[k_1, k_2, .., k_{n_r}]$.

Property 2. If the warehouse design problem has feasible solutions with n_r *racks, and the minimum total depth of the feasible solutions with* n_r *racks is* $D(n_r)$. *Then the feasible solutions with* $n_r + 1$ *racks should have a total depth of racks as* $\sum_{1 \le j \le n_r+1} k_j \le D(n_r)$ *to obtain a lower objective value.*

Therefore, the standard Branch & Bound algorithm can be modified as follows, and the performance of this modified B&B algorithm will be illustrated in section 4.6.

Step 1. Solve the MIP problem defined by Equation (4.6) and (4.19-4.30) with a fixed number of opening racks (i.e., n_r) where $n_r \in \{1,2, .., N_a\}$ while replacing the objective function (4.19) with (4.19') and N_a with n_r for all constraint sets. Record the summation of racks depth, $D(n_r)$, for

each n_r, and if the model is infeasible for a given number of racks, $D(n_r) = +\infty$. Moreover, we force $D(n_r) \leq D(n_r - 1)$, for $n_r \in \{2, .., N_a\}$. If $D(n_r) = +\infty$ for all $n_r \in \{1,2, .., N_a\}$, the warehouse design problem is infeasible and stops, otherwise; go to step 2.

$$\min \sum_{j=1}^{n_r} \sum_{k=1}^{K} k n_{ik}^{z} \qquad (4.19')$$

Step 2. Solve the product allocation problem with the feasible racks configurations obtained from step 1, choose the lowest objective value as the initial best known upper bound (i.e., UB), and set the initial best known lower bound (i.e., LB) of the warehouse design problem as zero. Create a list $q = [[1], [2], \ldots, [D]]$ where $D = \min (D(1), \max_{i \in I} Q_i)$.

Step 3. Loop until q is empty or $LB = UB$:

1. Take a node N off the queue following last-in-first-out.
2. If the rack configuration represented by node $N = [k_1, k_2, .., k_{n_r}]$ has a total depth larger than $D(n_r - 1)$, discard node N.
3. Else If the lower bound obtained by node N (i.e., lb) is larger than LB, set $LB = lb$.
4. If $lb \leq UB$ and the node N cannot provide an upper bound (i.e., ub), branch on node N and add the newly generated nodes into q.
5. Else if $lb < UB$, and node N can provide an upper bound that $ub < UB$, set $UB = ub$ and discard node N
6. Else if $lb > UB$, discard node N.

4.6 Numerical Experiments

The parameters of the SKUs' size (i.e., size of storage cells) and equipment speed in numerical experiments were taken from Dong et al. (2021). The width (x-direction; w), height (y-direction; h), and length (z-direction; l) of a storage cell are considered as $w = l = 1.4m$ and $h = 2m$. The velocity of shuttles (V_s) is $1.5m/s$. The velocity of cranes in x-direction and y-direction are $V_c^x = 2.5m/s \; and \; V_c^y = 0.5m/s$, respectively. With the given size of storage cells and equipment speed, simulations were first conducted to verify the expected cycle time models (i.e., Equation (4.1) and (4.2)).

4.6.1 Validation of Expected Cycle Time Models

Since the expected cycle time model for 2D AS/RS is taken directly from literature and has already been validated, here, we only have to verify the expected cycle time model for the 3D AS/RS rack. The 3D AS/RS rack considered for simulation is a real-world rack that has been analyzed by Dong et al. (2021). The rack has three layers, 80 columns, and each lane has 15 storage cells. The total number of storage lanes is 240, and the rack can store 3,600 SKUs. 10 scenarios are created in terms of the number of shuttles in the rack (i.e., m_j) which ranges from 24 to 240 with a step size of 24. For each scenario, we created 10 subcases, where the number of retrieval tasks is fixed as 2000 and the storage locations of the required SKUs are randomly selected. We define $|\Delta|$ as the average absolute difference between simulation results and cycle times calculated by the expected cycle time model (i.e., Equation (4.2)) over 10 subcases. Furthermore, σ represents the standard deviation of the simulation results across the 10 subcases. Table 4-1 lists both $|\Delta|$ and σ (in seconds) for different numbers of shuttles, and shows all $|\Delta|$ are much smaller than σ, which validates the accuracy of the expected cycle time model.

4.6.2 Compare B&B, The Modified B&B, and The MIP Solved by Gurobi

Numerical experiments were conducted to compare the performance of the standard B&B algorithm, the modified B&B algorithm, and solving the MIP defined by Equation (4.6) and (4.19-4.30) with Gurobi. The running time of Gurobi is limited as the maximum value of the running time of the standard B&B algorithm and 60 seconds. When it reaches the time limitation, Gurobi will stop and return a feasible solution if it can find one. 10 cases are created in terms of different numbers of SKU types (i.e., $|I|$), different numbers of storage lanes in one rack (i.e., n), and different inventory levels (i.e., Q_i) and demand rates (i.e., d_i) of each SKU type i for comparing these three solving methods.

The maximum number of racks allowed in a warehouse is assumed to be $N_a = 8$. The daily cost (e.g., hardware cost, equipment operation and maintenance cost, and the land cost for an aisle) of having one shuttle (C_s) and one crane (C_v) are considered as $\$0.2/day$ and $\$130/day$. The rack and warehouse space cost (e.g., land cost, rack cost, energy consumption) is $\$0.1/m^3/day$. The benefit loss for having an expected cycle time higher than the benchmark expected cycle time by

Table 4-1. Comparison between Simulation and Expected Cycle Time Model for 3D AS/RS

m_j	24	48	72	96	120	144	168	192	216	240
$\|\Delta\|$	0.53	0.53	0.50	0.44	0.34	0.44	0.40	0.33	0.33	0.22
σ	0.70	0.60	0.60	0.67	0.71	0.78	0.61	0.54	0.53	0.54

one second (i.e., C_t) is randomly picked from $\$10/second/day$ to $\$130/second/day$. The details for estimating the cost parameters can be found in Appendix H.

The performance of each solving method under these 10 cases are illustrated in Table 4-2. The objective value ($/day) obtained by the standard B&B algorithm, the modified $B\&B$ algorithm, and solving the MIP defined by Equation (4.6) and (4.19-4.30) are denoted by $C_{B\&B}, C_{B\&B'}$, and C_{MIP}. The running time (in seconds) of each solving method is shown by $T_{B\&B}, T_{B\&B'}$, and T_{MIP}, respectively. Moreover, $NON_{B\&B}$ and $NON_{B\&B'}$ demonstrate the number of nodes in the branch-and-bound tree generated by the standard B&B algorithm and the modified B&B algorithm, respectively. In addition, we report the gap (i.e., G_{MIP}) of Gurobi when it reaches the time limitation to indicate if the optimal solution is found.

As demonstrated by Table 4-2, the standard B&B procedure can always reach the optimal solution since it is an exact algorithm. Moreover, the standard B&B algorithm can usually spend much less time to reach the optimal solution than the MIP. The modified B&B algorithm can always get the same solution as the standard B&B algorithm and can usually generate a much smaller branch-and-bound tree, and therefore, requires a much shorter time to reach the optimal solution (e.g., $(n, |I|, \sum_{i \in N} Q_i, C_t) = (28,27{,}675,120)$). Another 50 cases were also created for further comparing the standard and the modified B&B algorithm and the results are shown in Appendix K. Surprisingly, the modified B&B algorithm can always reach the optimal solution within a shorter running time than the standard B&B algorithm. Furthermore, the solving time of the warehouse design problem is much shorter when C_t is low or $n > |I|$. That's because when C_t is low or $n > |I|$, the warehouse can have a smaller number of racks which helps to reduce the solution space (i.e., branch-and-bound tree). In addition, even not shown in this paper, the solving time is also affected by the distribution of the demand rates and inventory levels across different SKU types.

4.6.3 Sensitivity Analysis of the Warehouse Design Problem

More numerical experiments were conducted for large case problems solved by the modified B&B algorithm to analyze the impacts of cost parameters and the distribution of inventory levels and demand rates across different SKU types on the warehouse design. To generate the demand rate (i.e., d_i) for each SKU type, we used the method proposed by Hausman et al. (1976) to calculate the normalized demand rate for each SKU type i.

Table 4-2. Comparison of Solving Methods

$(n, \lvert I\rvert, \sum_{i\in N} Q_i, C_t)$	$C_{B\&B}$	$C_{B\&B'}$	C_{MIP}	$T_{B\&B}$	$T_{B\&B'}$	T_{MIP}	$NON_{B\&B}$	$NON_{B\&B'}$	C_{MIP}
$(10, 9, 90, 51)$	868.9	868.9	868.9	5.7	3.2	42	452	413	0%
(10,7,80,27)	599.5	599.5	599.5	3.8	0.7	2.8	114	69	0%
(10,9,116,40)	705.7	705.7	705.7	3.6	0.24	2.2	53	38	0%
(20,14,131,59)	736.3	736.3	736.3	3.8	1.03	8.4	90	63	0%
(20,16,172,69)	1133	1133	1159	32	5	60	878	176	25%
(14,17,196,118)	1528.7	1528.7	1528.7	17.7	4.5	20.8	444	92	0%
$(24, 27, 540, 120)$	2229.9	2229.9	2243.7	314	315	314	2321	2110	34%
(33,27,540,27)	849.4	849.4	858.4	79	53	80	709	532	2%
(28,27,675,120)	2384.8	2384.8	2398.1	15332	1527	15333	23671	7130	35%
(25,27,675,37)	904.7	904.7	914.5	407	128	407	1860	1118	1.4%

$$G(l) = l^v, \qquad 0 < l \leq 1 \qquad (4.31)$$

where $0 < v \leq 1$. Formulation (4.31) represents the ranked cumulative $(l^v \times 100)\%$ demand versus $(l \times 100)\%$ of SKU types. The value of v determines the distribution of demand rates across different SKU types. A greater v means a more homogenous demand rate across SKU types. For a given v, d_i is obtained through equations (4.32) and (4.33) (Hausman et al. 1976).

$$d_i = G\left(\frac{i}{N}\right) - G\left(\frac{i-1}{N}\right) \qquad i = 1, \dots, N \qquad (4.32)$$

$$G(0) = 0 \qquad (4.33)$$

Here, we consider four demand rate distribution scenarios, featured by $G(0.2) \in \{0.2, 0.4, 0.6, 0.8\}$. Higher $G(0.2)$ presents the demand rates across SKU types are more heterogeneous. $G(0.2) = 0.2$ (i.e., $v = 1$) implies that all SKU types have the same demand rate. The inventory levels are generated following a Gaussian distribution (i.e., $\mathcal{N}(\mu, \sigma^2)$), where $\sum_{i \in N} Q_i = 675$ units, μ =25 units, and the standard derivation (σ) ranges from 1 to 21 with a step size of 5 units. Moreover, we assume the SKU type i with a higher d_i corresponds to a higher inventory level, Q_i. The number of SKU types (i.e., $|I|$) under consideration is 27. The maximum number of racks allowed is 10, and each rack, if designed to be open, has 29 storage lanes. For each combination of demand rate and inventory level distributions, the daily cost of having an expected cycle time higher than the benchmark expected cycle time by one second is taken from the set [$\$1/second/day, \$25/second/day, \$50/second/day, \$75/second/day, \$100/second/day, \$125/second/day$] while the values of the other cost parameters are fixed.

Table 4-3 demonstrates the warehouse design problem under different values of C_t with $v = 0.4$ and $\sigma = 16$, where EC demonstrates the expected cycle time for retrieving an SKU under the optimal warehouse design, $[k_1, k_2, .., k_{n_r}]$ represents the optimal configuration (i.e., number and depth of racks), $[m_1, m_2, .., m_{n_r}]$ shows the number of shuttles on each rack, S shows the storage capacity (i.e., number of storage cells) of the system, and Avg and std denote the mean and standard derivation of racks depth.

As illustrated by Table 4-3, when the cost for not meeting the desired throughput capacity is low (i.e., $C_t = 1$), the warehouse design tends to have a fewer number but deep racks and one shuttle for each rack. Even though such design can lead to a high expected cycle time (i.e., 25.1 seconds)

Table 4-3. Sensitivity Analysis of C_t when $\boldsymbol{v} = 0.4$ and $\sigma = 16$

C_t	$C_{B\&B}$	EC	$[k_1, k_2, .., k_{n_r}]$	$[m_1, m_2, .., m_{n_r}]$	S	Avg	std
1	612	25.1	[9, 16]	[1, 1]	725	12.5	3.5
25	1104	13.9	[7, 8, 9]	[29, 29, 29]	696	8.0	0.8
50	1454	13.9	[7, 8, 9]	[29, 29, 29]	696	8.0	0.8
75	1750	11.3	[5, 6, 6, 7]	[29, 29, 29, 29]	696	6.0	0.7
100	1921	11.3	[5, 6, 6, 7]	[29, 29, 29, 29]	696	6.0	0.7
125	2177	9.8	[3, 5, 5, 5, 6]	[29, 29, 29, 29, 29]	696	4.8	0.9

and wasted storage space (i.e., $S = 725$, while $\sum_{i \in N} Q_i = 675$), having fewer racks can reduce the number of cranes, which is much expensive than C_t and C_{rw}. By increasing C_t from one to 25, the warehouse design tends to have a larger number but shallower racks ($[k_1, k_2, .., k_{n_r}]$ and Avg) and the depth of each rack is more evenly distributed (std) than the design under $C_t = 1$, and we do expect a few 2D AS/RS racks if C_t gets extremely high. Moreover, when $C_t = 25$, even though the depth of each rack is reduced, each rack has shuttles for each lane to reduce the expected cycle time, which is consistent with Theorem 4.2 that a 3D AS/RS rack should either have one or n shuttles. That's because when C_t is high, shallow racks and more shuttles are needed to reduce the expected cycle time. However, by increasing C_t from 25 to 50 or from 50 to 25, the warehouse design does not change, which indicates that the marginal impact of increasing C_t on warehouse design decrease in C_t. In addition, even though we did not run sensitivity analysis on C_v, C_{rw}, and C_s, it is straightforward that decreasing crane's cost, C_v, will have the same impact as increasing C_t. Increasing C_{rw} will make the warehouse have more racks, since, as demonstrated in Table 4-3, by increasing the number of racks, the total storage space can be reduced, and will also reduce the expected cycle time. However, for any given values of $\sum_{i \in N} Q_i$ and n, there is a lower bound for the storage capacity, which is $\left\lceil \frac{\sum_{i \in N} Q_i}{n} \right\rceil n$ and increasing C_{rw} will not have any impact on warehouse design once the storage capacity reaches the lower bound. If the shuttles become more expensive (i.e., high C_s), the system needs to reduce the number of shuttles on each rack. In that case, if C_t is small, the system may just simply reduced the number of shuttles without changing the configuration of racks. Otherwise, if C_t is large, the system will need a larger number but shallower racks to reduce the requirement on shuttles without hurting the expected cycle time.

Table 4-4 demonstrates the impacts of the variations of inventory levels and demand rates across SKU types on warehouse design. Table 4-4 only shows the warehouse design under different distributions of inventory levels and demand rates when $C_t = 75$. The results under the other values of C_t follow the same pattern and can be found in Appendix L. As shown in Table 4-4, when $G(0.2)$ is low (e.g., $G(0.2) = 0.2$), which means a more homogeneous demand rate distribution across SKU types, heterogeneous distribution of inventory levels across SKU types (high σ) will lead to a warehouse design where the racks are unevenly distributed in terms of depth (i.e., high std). That's because when σ is high, the storage capacity requirement will be

Table 4-4. Sensitivity Analysis on Inventory Levels and Demand Rates Distribution when $C_t = 75$

$G(0.2)$	σ	$C_{B\&B}$	EC	$[k_1, k_2, .., k_{n_r}]$	$[m_1, m_2, .., m_{n_r}]$	S	Avg	std
0.2	1	1765	11.5	[5, 6, 6, 7]	[29, 29, 29, 29]	696	6.0	0.71
	6	1757	11.4	[5, 6, 6, 7]	[29, 29, 29, 29]	696	6.0	0.71
	11	1739	11.2	[4, 5, 7, 8]	[29, 29, 29, 29]	696	6.0	1.58
	16	1720	10.9	[4, 5, 7, 8]	[29, 29, 29, 29]	696	6.0	1.58
	21	1691	10.5	[3, 5, 7, 9]	[29, 29, 29, 29]	696	6.0	2.2
0.4	1	1739	11.2	[4, 6, 6, 8]	[29, 29, 29, 29]	696	6.0	1.41
	6	1751	11.3	[5, 6, 6, 7]	[29, 29, 29, 29]	696	6.0	0.71
	11	1750	11.3	[5, 6, 6, 7]	[29, 29, 29, 29]	696	6.0	0.71
	16	1751	11.3	[5, 6, 6, 7]	[29, 29, 29, 29]	696	6.0	0.71
	21	1746	11.2	[4, 6, 7, 7]	[29, 29, 29, 29]	696	6.0	1.2
0.6	1	1662	10.1	[3, 6, 7, 8]	[29, 29, 29, 29]	696	6.0	1.87
	6	1683	10.4	[3, 6, 7, 8]	[29, 29, 29, 29]	696	6.0	1.87
	11	1694	10.6	[3, 6, 7, 8]	[29, 29, 29, 29]	696	6.0	1.87
	16	1706	10.7	[3, 6, 7, 8]	[29, 29, 29, 29]	696	6.0	1.87
	21	1718	10.9	[4, 6, 6, 8]	[29, 29, 29, 29]	696	6.0	1.41
0.8	1	1528	10.3	[3, 8, 13]	[29, 29, 29, 29]	696	8.0	4.0
	6	1554	10.7	[3, 9, 12]	[29, 29, 29, 29]	696	8.0	3.7
	11	1563	10.8	[2, 9, 13]	[29, 29, 29, 29]	696	8.0	4.5
	16	1593	11.2	[3, 9, 12]	[29, 29, 29, 29]	696	8.0	3.7
	21	1623	9.6	[3, 6, 7, 8]	[29, 29, 29, 29]	696	6.0	1.87

mainly from a few SKU types and the inventory levels for the other SKUs types are low, which motivates the warehouse to have shallower racks for SKU types with low inventory level, and open deep racks for accommodating SKU types with the majority of inventories to reduce wasted storage spaces. However, with the increase of $G(0.2)$, the SKU types with most of the inventories have higher demand rates than the other SKU types, and recall the expression of the cost of not meeting the benchmark expected cycle time (i.e., $\sum_{i\in I}\sum_{j=1}^{N_a} d_i \frac{y_{ij}}{Q_i} C_t EC_j$), the SKU types with high demand rates need to be accommodated into shallower racks which against the impact of increasing σ. Similarly, we expect that increasing C_t will also reduce the impact of increasing σ. Therefore, when $G(0.2)$ or C_t is high, the distribution of inventory levels of SKU types will not have a significant impact on warehouse design. Moreover, when σ is fixed, having a more heterogeneous distribution of demand rate will lead to a warehouse design with a high variation of depth of each rack. This can also be explained by looking at $\sum_{i\in I}\sum_{j=1}^{N_a} d_i \frac{y_{ij}}{Q_i} C_t EC_j$. However, when σ is high, the impact of increasing $G(0.2)$ will not be as significant as its impact under lower σ, due to the contradictory impact of the distributions of inventory levels and demand rates on warehouse design.

Moreover, even though 2D AS/RS racks lead to a much smaller expected cycle time than 3D racks, we observed that 2D AS/RS racks have rarely been used. This is because of the high cost of having a crane in the system, but if C_t is extremely high, we do expect the design of 2D racks. However, it demonstrates that 3D AS/RS is usually better than 2D AS/RS, which is consistent with the fact 2D AS/RS is getting replaced by other AS/RS options practice (e.g., 3D AS/RS).

4.7 Conclusion and Future Research

This study considers the warehouse design problem by allowing the mixed-use of 2D AS/RS and 3D AS/RS. The objective is to minimize warehouse cost while maintaining a certain level of throughput capacity measured by expected cycle time under different business needs characterized by the variations of inventory levels and demand rates across different SKU types. The warehouse design decisions include the number and dimension of racks, the number of equipment, and the SKU-to-system assignment. The expected cycle model for 3D AS/RS with shuttle-based DMMS has never been done and the gap is filled by this study. The warehouse design problem is first modeled as an MINP and then converted to MIP based on two optimality

conditions. A branch-and-bound algorithm is developed and then modified based on our observation of numerical experiments to further reduce the algorithm's running time. Numerical experiments showed that the modified B&B algorithm can usually reach the optimal solution within a much shorter time than the original B&B algorithm.

Numerical experiments are conducted to analyze the impacts of cost parameters and the distribution of inventory levels and demand rates across different SKU types on warehouse design.

- When C_t is low (i.e., $C_t = 1$), warehouse design tends to have a fewer number but deep racks and fewer shuttles in the system to reduce the equipment cost. When C_t increases, a warehouse needs a larger number but shallower racks with more evenly distributed racks depth to reduce the expected cycle time for retrieving an SKU. The marginal impact of increasing C_t on warehouse design decreases in C_t. Increasing C_{rw} and C_s or reducing C_v have a similar impact to increasing C_t.
- When the demand rates across SKU types are homogeneous, a heterogeneous distribution of inventory levels will lead to a warehouse design where the depth of each rack is more unevenly distributed (i.e., high std).
- Having a heterogeneous distribution of demand rates will also make the warehouse design with a more unevenly distributed depth of racks.
- If a warehouse has a design as $[k_1, k_2, .., k_{n_r}]$ where n_r is the number of racks and k_j is the depth of rack j, only when $\sum_{1 \le j \le n_r+1} k'_j \le \sum_{1 \le j \le n_r} k_j$, the system can consider a new design as $[k'_1, k'_2, .., k'_{n_r}, k'_{n_r+1}]$.
- If n_r racks can provide enough storage capacity and the minimum total depth of n_r racks is $D(n_r)$. The warehouse can consider a new design $[k_1, k_2, .., k_{n_r+1}]$ only when $\sum_{1 \le j \le n_r+1} k_j \le D(n_r)$.
- 3D AS/RS is better than 2D AS/RS which is consistent with the industrial trends.

Moreover, based on the optimality conditions,

- Each rack should try to have a square-in-time picking face.
- A 3D AS/RS rack should either have one or n shuttles, where n is the number of storage lanes on a rack.

In the B&B algorithms, solving the product allocation problem with given racks configurations by Gurobi is used to obtain the upper bounds. However, when the problem size is too large (in terms of SKU types and inventory) or the distribution of inventory levels and demand rates across SKU types is extremely heterogeneous, the solver still needs a large amount of running time. An efficient algorithm for the product allocation problem will be very helpful to reduce the running time of the branch-and-bound algorithms. Moreover, this study only considered the crane-based 2D AS/RS and crane-based 3D AS/RS as candidates. However, more than 20 variants of AS/RS exist and future studies incorporating all of these variants will be necessary to facilitate warehouse design. Furthermore, this study only included the number and dimensions of racks, the number of equipment, and product allocation as decision variables, and only used the cost and expected cycle time as performance metrics. However, a system's performance can also be measured from many different perspectives (e.g., energy consumption, transaction waiting time), and the system's performance can be affected by many other decision factors (e.g., number and locations of I/O points). A comprehensive tool considering all AS/RS options, performance metrics, and decision factors is necessary. However, given the fact that the warehouse design problem considered in this study is already difficult enough, a study that considers all decision factors and systems will be extremely complex. Therefore, based on various optimization models and algorithms that are developed for each type of AS/RS, statistical analyses and sensitivity analyses could be adopted to numerically build functions representing the relationship between design factors and performance metrics (i.e., mechanisms by which different design parameters affect system performance under given environments). Those functions should be used by practitioners to quickly determine if AS/RS is applicable and, if so, how to design and operate that system.

CHAPTER FIVE
CONCLUSION AND FUTURE DIRECTIONS

This study aims to create a tool for facilitating the warehouse design problem considering multiple AS/RS systems. Chapter 2 and chapter 3 focused on the operation and performance of crane-based 3D AS/RS and SBS/RS, respectively. More particularly, in chapter 2, we considered how to sequence a group of retrieval requests in a crane-based 3D AS/RS with shuttle-based DMMs to minimize the makespan. A mixed-integer programming model was developed to represent the problem, and the problem was proven to be NP-hard. Based on Theorem 2.1, a heuristic was developed to solve the problem quickly. Numerical experiments have demonstrated that the developed heuristic is better than FCFS, which is widely used in practice; the PPS-SL, which is the best heuristic for task scheduling in 3D AS/RS from literature; and the genetic algorithm which is widely adopted for task scheduling problems, in terms of solving time and solution quality. Management insights have been summarized through numerical experiments: 1) When the number of retrieval tasks is small (e.g., when a short planning horizon is adopted for high responsiveness), having more shuttles can improve the system performance, but 2) When there are many tasks to schedule, for example, in a situation with a long planning horizon, using a higher crane speed rather than adding more shuttle can improve system efficiency (reduce makespan) more.

The impacts of different operation modes (i.e., SC operations and DC operations), storage policies (i.e., random storage and class-based storage), and shuttle dispatching rules (i.e., random, distance-based, and demand-rate based shuttle dispatching) on the performance of a tier-to-tier SBS/RS have been analyzed in Chapter 3. The system was modeled as a DTMC to calculate shuttles distribution under different operational policies. Based on the shuttle distribution, expected travel time models were developed and validated through simulation. Through numerical experiments, we observed: 1) for each operation mode, the demand distribution information is important for selecting the appropriate storage policy and shuttle dispatching rule; 2) class-based storage is always better than random storage while the best dispatching rule is affected by demand distribution patterns, the number of shuttles, and operational cycle (i.e., SC and DC); 3) when the demand rate across classes is more homogeneous, the number of shuttles in a system is low, and the system operates in SC cycles, the demand rate-based shuttle dispatching rule is the best; 4) when the demand rate across classes is more homogeneous, the number of shuttles in a system is low, and the system operates in DC cycles, the other two shuttle dispatching rules might be better choices; 5) If demands are more heterogeneous, the demand

rate-based shuttle dispatching rule is competitive in every scenario, while the advantage of this rule will increases first and then decrease in the number of shuttles; 6) when random storage policy is applied, the distance-based shuttle dispatching rule is always better than random dispatching rule under SC operations, but under DC operations, depending on the equipment's speed, random dispatching rule can be better.

Based on the observations from chapter 2 and chapter 3, in chapter 4, the warehouse design problem considering 2D AS and 3D AS/RS as technology candidates was modeled and solved. The warehouse design problem considered the trade-off between warehouse investment and the system's throughput capacity with the number and dimensions of AS/RS racks, the number of equipment, and the product allocation as decision factors. The warehouse design problem was first modeled as an MINP and then converted to MIP based on two optimality conditions. A branch-and-bound algorithm was developed and then modified based on our observation of numerical experiments to further reduce the algorithm's running time. Numerical experiments showed that the modified B&B algorithm can usually reach the optimal solution within a much shorter time than the original B&B algorithm. When focusing on reducing the investment, a warehouse should have a fewer number but deep racks and fewer shuttles in the system to reduce the equipment cost. If the objective is to improve the throughput capacity, a warehouse needs a larger number but shallower racks with more evenly distributed racks depth to reduce the expected cycle time for retrieving an SKU. When the demand rates across SKU types are homogeneous, a heterogeneous distribution of inventory levels will lead to a warehouse design where the depth of each rack is more unevenly distributed. Having a heterogeneous distribution of demand rates will also make the warehouse design with a more unevenly distributed depth of racks. Surprisingly, the warehouse design problem can be divided into two steps: 1) solve the warehouse design problem to minimize the total depth of racks iteratively on the different numbers of racks; 2) solve the product allocation with all feasible solutions from step one and select the one with the minimum objective value.

However, this work only covers the crane-based 2D AS/RS and crane-based 3D AS/RS as candidates and more than 20 variants of AS/RS exist and future studies incorporating all of these variants will be necessary to facilitate warehouse design. Furthermore, this study only included the number and dimensions of racks, the number of equipment, and product allocation as decision variables, and only used the cost and expected cycle time as performance metrics. However, a

system's performance can also be measured from many different perspectives (e.g., energy consumption, transaction waiting time), and the system's performance can be affected by many other decision factors (e.g., number and locations of I/O points). A comprehensive tool considering all AS/RS options, performance metrics, and decision factors is necessary. However, given the fact that the warehouse design problem considered in this study is already difficult enough, a study that considers all decision factors and systems will be extremely complex. Therefore, based on various optimization models and algorithms that are developed for each type of AS/RS, statistical analyses and sensitivity analyses could be adopted to numerically build functions representing the relationship between design factors and performance metrics (i.e., mechanisms by which different design parameters affect system performance under given environments). Therefore, future studies should focus on the establishment of relationship functions f_i^j so that $Y^j = f_i^j(X)$, where X is the vector representing design parameters and Y^j is the vector of the performance metrics for AS/RS variant j. Furthermore, optimization models should be established to $U_k^j = \max_{X \in \Omega_k} g_k(Y^j) = f_i^j(X)$, where g_k and Ω_k represent the utility preference and constraints (e.g., responsiveness requirement, order variety, order sizes) of a given user k, respectively; and U_k^j is the utility of user k for variant j. Knowing U_k^j, the practitioners can quickly determine if an AS/RS is applicable and, if so, how to design and operate that system.

LIST OF REFERENCES

Alibaba.com, 2021a. Stacker Crane Designed for High Bay Warehouse Automated Storage and Retrieval. URL https://www.alibaba.com/product-detail/Stacker-Crane-Stacker-Crane-Designed-For_62360145937.html?spm=a2700.galleryofferlist.normal_offer.d_title.62182d51989CCu&s=p

Alibaba.com, 2021b. Industrial Warehouse Cold Storage Automatic Pallet Runner Radio Shuttle System. URL https://www.alibaba.com/product-detail/Pallet-Shuttle-Industrial-Pallet-Industrial-Warehouse_1600089518412.html?spm=a2700.galleryofferlist.normal_offer.d_title.7b041239bjYdXB&s=p

Asokan, P., Jerald, J., Arunachalam, S., & Page, T., 2008. Application of adaptive genetic algorithm and particle swarm optimisation in scheduling of jobs and AS/RS in FMS. Int. J. Manuf. Res. 3, 393–405.

Azadeh, K., De Koster, R., Roy, D., 2019a. Robotized and automated warehouse systems: Review and recent developments. Transp. Sci. 53, 917–945. https://doi.org/10.1287/trsc.2018.0873

Azadeh, K., Roy, D., De Koster, R., 2019b. Design, modeling, and analysis of vertical robotic storage and retrieval systems. Transp. Sci. 53, 1213–1234. https://doi.org/10.1287/trsc.2018.0883

Baker, P., Canessa, M., 2009. Warehouse design : A structured approach. Eur. J. Oper. Res. 193, 425–436. https://doi.org/10.1016/j.ejor.2007.11.045

Bastian Solutions, 2010. Options Available with ASRS Technology. URL https://www.bastiansolutions.com/blog/options-available-with-asrs-technology/ (accessed 5.26.21).

Battini, D., Boysen, N., Emde, S., 2013. Just-in-Time supermarkets for part supply in the automobile industry. J Manag Control 24, 209–217. https://doi.org/10.1007/s00187-012-0154-y

Bessenouci, H.N., Sari, Z., Ghomri, L., 2012. Metaheuristic based control of a flow rack automated storage retrieval system. J. Intell. Manuf. 23, 1157–1166.

https://doi.org/10.1007/s10845-010-0432-1

Bortolini, M., Accorsi, R., Gamberi, M., Manzini, R., Regattieri, A., 2015. Optimal design of AS/RS storage systems with three-class-based assignment strategy under single and dual command operations. Int. J. Adv. Manuf. Technol. 79, 1747–1759. https://doi.org/10.1007/s00170-015-6872-1

Bottani, E., Volpi, A., Montanari, R., 2019. Design and optimization of order picking systems: An integrated procedure and two case studies. Comput. Ind. Eng. 137, 106035. https://doi.org/10.1016/j.cie.2019.106035

Boysen, N., De Koster, R., Weidinger, F., 2019. Warehousing in the e-commerce era: A survey. Eur. J. Oper. Res. 277, 396–411. https://doi.org/10.1016/j.ejor.2018.08.023

Boysen, N., Emde, S., Hoeck, M., Kauderer, M., 2015. Part logistics in the automotive industry: Decision problems, literature review and research agenda. Eur. J. Oper. Res. 242, 107–120. https://doi.org/10.1016/j.ejor.2014.09.065

Boysen, N., Stephan, K., 2016. A survey on single crane scheduling in automated storage/retrieval systems. Eur. J. Oper. Res. https://doi.org/10.1016/j.ejor.2016.04.008

Bozer, Y.A., White, J.A., 1990. Design and performance models for end-of-aisle order picking systems. Manage. Sci. 36, 852–866. https://doi.org/10.1287/mnsc.36.7.852

Bozer, Y.A., White, J.A., 1984. Travel-time models for automated storage/retrieval systems. IIE Trans. 16, 329–338. https://doi.org/10.1080/07408178408975252

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 Trans. 46, 905–927. https://doi.org/10.1080/0740817X.2013.849832

Cao, W., Zhang, M., 2017. The optimization and scheduling research of shuttle combined vehicles in automated automatic three-dimensional warehouse. Procedia Eng. 174, 579–587. https://doi.org/10.1016/j.proeng.2017.01.190

Carlo, H.J., Vis, I.F.A., 2012. Sequencing dynamic storage systems with multiple lifts and shuttles. Int. J. Prod. Econ. 140, 844–853. https://doi.org/10.1016/j.ijpe.2012.06.035

Chang, S.H., Egbelu, P.J., 1997. Relative pre-positioning of storage/retrieval machines in

automated storage/retrieval systems to minimize maximum system response time. IIE Trans. 29, 303–312. https://doi.org/10.1080/07408179708966336

Changpeng Shen, Yaohua Wu, Danyu Zhang, 2010. A selection method of manual and semi-automated order picking systems based on filling curve and time model, in: 2010 IEEE International Conference on Automation and Logistics. IEEE, pp. 169–176. https://doi.org/10.1109/ICAL.2010.5585274

Chetty, O.V.K., Reddy, M.S., 2003. Genetic algorithms for studies on AS/RS integrated with machines. Int. J. Adv. Manuf. Technol. 22, 932–940. https://doi.org/10.1007/s00170-003-1629-7

Coombs, C., 2018. Amazon's warehouse expansion helps drives up industrial land prices - Puget Sound Business Journal. URL https://www.bizjournals.com/seattle/news/2018/01/03/amazon-s-warehouse-expansion-helps-drives-up.html (accessed 5.26.21).

D'Antonio, G., Chiabert, P., 2019. Analytical models for cycle time and throughput evaluation of multi-shuttle deep-lane AVS/RS. Int. J. Adv. Manuf. Technol. 104, 1919–1936. https://doi.org/10.1007/s00170-019-03985-8

D'Antonio, G., Maddis, M. De, Bedolla, J.S., Chiabert, P., Lombardi, F., 2018. Analytical models for the evaluation of deep-lane autonomous vehicle storage and retrieval system performance. Int. J. Adv. Manuf. Technol. 94, 1811–1824. https://doi.org/10.1007/s00170-017-0313-2

Davarzani, H., Norrman, A., 2015. Toward a relevant agenda for warehousing research : literature review and practitioners ' input. Logist. Res. https://doi.org/10.1007/s12159-014-0120-1

De Koster, R., Johnson, A.L., Roy, D., 2017. Warehouse design and management. Int. J. Prod. Res. 55, 6327–6330. https://doi.org/10.1080/00207543.2017.1371856

De Koster, R., Le-Duc, T., Yugang, Y., 2008. Optimal storage rack design for a 3-dimensional compact AS/RS. Int. J. Prod. Res. 46, 1495–1514. https://doi.org/10.1080/00207540600957795

Dong, W., Jin, M., Wang, Y., Kelle, P., 2021. Retrieval scheduling in crane-based 3D automated retrieval and storage systems with shuttles. Ann. Oper. Res. https://doi.org/10.1007/s10479-

021-03967-8

Dooly, D.R., Lee, H.F., 2008. A shift-based sequencing method for twin-shuttle automated storage and retrieval systems. IIE Trans. 40, 586–594. https://doi.org/10.1080/07408170701730776

Eder, M., 2020. Analytical model to estimate the performance of shuttle-based storage and retrieval systems with class-based storage policy. Int. J. Adv. Manuf. Technol. 107, 2091–2106. https://doi.org/10.1007/s00170-020-04990-y

Eder, M., 2019. An analytical approach for a performance calculation of shuttle-based storage and retrieval systems. Prod. Manuf. Res. 7, 255–270. https://doi.org/10.1080/21693277.2019.1619102

Eder, M., Kartnig, G., 2016. Throughput analysis of S/R shuttle systems and ideal geometry for high performance. FME Trans. 44, 174–179. https://doi.org/10.5937/fmet1602174E

Ekren, B.Y., 2020. A simulation-based experimental design for SBS/RS warehouse design by considering energy related performance metrics. Simul. Model. Pract. Theory 98. https://doi.org/10.1016/j.simpat.2019.101991

Ekren, B.Y., 2017. Graph-based solution for performance evaluation of shuttle-based storage and retrieval system. Int. J. Prod. Res. 55, 6516–6526. https://doi.org/10.1080/00207543.2016.1203076

Ekren, B.Y., Akpunar, A., 2021. An open queuing network-based tool for performance estimations in a shuttle-based storage and retrieval system. Appl. Math. Model. 89, 1678–1695. https://doi.org/10.1016/j.apm.2020.07.055

Ekren, B.Y., Akpunar, A., Sari, Z., Lerher, T., 2018. A tool for time, variance and energy related performance estimations in a shuttle-based storage and retrieval system. Appl. Math. Model. 63, 109–127. https://doi.org/10.1016/j.apm.2018.06.037

Ekren, B.Y., Heragu, S.S., 2012. Performance comparison of two material handling systems: AVS/RS and CBAS/RS. Int. J. Prod. Res. 50, 4061–4074. https://doi.org/10.1080/00207543.2011.588627

Ekren, B.Y., Heragu, S.S., 2010. Simulation-based regression analysis for the rack configuration

of an autonomous vehicle storage and retrieval system. Int. J. Prod. Res. 48, 6257–6274. https://doi.org/10.1080/00207540903321665

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 Trans. Autom. Sci. Eng. 10, 205–215. https://doi.org/10.1109/TASE.2012.2200676

Ekren, B.Y., Sari, Z., Lerher, T., 2015. Warehouse design under class-based storage policy of shuttle-based storage and retrieval system. IFAC-PapersOnLine 28, 1152–1154. https://doi.org/10.1016/j.ifacol.2015.06.239

Fang, Y., Tang, M., 2014. The AVS/RS Scheduling Optimization Based on Improved AFSA. Int. J. Control Autom. 7, 53–64. https://doi.org/10.14257/ijca.2014.7.10.06

Fukunari, M., Malmborg, C.J., 2009. A network queuing approach for evaluation of performance measures in autonomous vehicle storage and retrieval systems. Eur. J. Oper. Res. 193, 152–167. https://doi.org/10.1016/j.ejor.2007.10.049

Fukunari, M., Malmborg, C.J., 2008. An efficient cycle time model for autonomous vehicle storage and retrieval systems. Int. J. Prod. Res. 46, 3167–3184. https://doi.org/10.1080/00207540601118454

Gagliardi, J.P., Renaud, J., Ruiz, A., 2012. Models for automated storage and retrieval systems: A literature review. Int. J. Prod. Res. 50, 7110–7125. https://doi.org/10.1080/00207543.2011.633234

Gaku, R., Takakuwa, S., 2018. Simulation analysis of large-scale shuttle vehicle-type mini-load AS/RS systems, in: 2018 Winter Simulation Conference (WSC). IEEE, pp. 2966–2976. https://doi.org/10.1109/WSC.2018.8632394

Gu, J., Goetschalckx, M., McGinnis, L.F., 2010. Research on warehouse design and performance evaluation: A comprehensive review. Eur. J. Oper. Res. 203, 539–549. https://doi.org/10.1016/j.ejor.2009.07.031

Guo, X., Yu, Y., De Koster, R., 2016. Impact of required storage space on storage policy performance in a unit-load warehouse. Int. J. Prod. Res. 54, 2405–2418. https://doi.org/10.1080/00207543.2015.1083624

Ha, Y., Chae, J., 2019. A decision model to determine the number of shuttles in a tier-to-tier SBS/RS. Int. J. Prod. Res. 57, 963–984. https://doi.org/10.1080/00207543.2018.1476787

Ha, Y., Chae, J., 2018. Free balancing for a shuttle-based storage and retrieval system. Simul. Model. Pract. Theory 82, 12–31. https://doi.org/10.1016/j.simpat.2017.12.006

Hamzaoui, M.A., Arbaoui, T., Yalaoui, F., Sari, Z., 2021. An exact optimization method based on dominance properties for the design of AS/RSs. Transp. Res. Part E Logist. Transp. Rev. 146, 102204. https://doi.org/10.1016/j.tre.2020.102204

Han, M.-H., McGinnis, L.F., Shieh, J.S., White, J.A., 1987. On sequencing retrievals in an automated storage/retrieval system. IIE Trans. 19, 56–66. https://doi.org/10.1080/07408178708975370

Hausman, W.H., Schwarz, L.B., Graves, S.C., 1976. Optimal storage assignment in automatic warehousing systems. Manage. Sci. 22, 629–638. https://doi.org/10.1287/mnsc.22.6.629

Heragu, S.S., Cai, X., Krishnamurthy, A., Malmborg, C.J., 2011. Analytical models for analysis of automated warehouse material handling systems. Int. J. Prod. Res. 49, 6833–6861. https://doi.org/10.1080/00207543.2010.518994

Jaghbeer, Y., Hanson, R., Johansson, M.I., 2020. Automated order picking systems and the links between design and performance: a systematic literature review. Int. J. Prod. Res. 2020, 1–17. https://doi.org/10.1080/00207543.2020.1788734

Korte, B., Vygen, J., 2012. Bin-Packing, in: Kombinatorische Optimierung. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 499–516. https://doi.org/10.1007/978-3-642-25401-7_18

Kriehn, T., Schloz, F., Wehking, K.H., Fittinghoff, M., 2018. Impact of class-based storage, sequencing of retrieval requests and warehouse reorganisation on throughput of shuttle-based storage and retrieval systems. FME Trans. 46, 320–329. https://doi.org/10.5937/fmet1803320K

Kriehn, T., Schulz, R., Fittinghoff, M., 2019. Algorithm and analytical model to optimize class-based storage of shuttle-based storage and retrieval systems, in: Proceedings of the XXIII International Conference MHCL 2019. FME Belgrade, pp. 209–216.

Küçükyaşar, M., Ekren, B.Y., Lerher, T., 2021. Cost and performance comparison for tier-captive and tier-to-tier SBS/RS warehouse configurations. Int. Trans. Oper. Res. 28, 1847–1863. https://doi.org/10.1111/itor.12864

Kumar, S., Narkhede, B.E., Jain, K., 2021. Revisiting the warehouse research through an evolutionary lens: a review from 1990 to 2019. Int. J. Prod. Res. https://doi.org/10.1080/00207543.2020.1867923

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. Appl. Math. Model. 31, 2332–2346. https://doi.org/10.1016/j.apm.2006.09.011

Lee, H.F., 1997. Performance analysis for automated storage and retrieval systems. IIE Trans. 29, 15–28. https://doi.org/10.1080/07408179708966308

Lerher, T., 2016. Travel time model for double-deep shuttle-based storage and retrieval systems. Int. J. Prod. Res. 54, 2519–2540. https://doi.org/10.1080/00207543.2015.1061717

Lerher, T., Ekren, B.Y., Dukic, G., Rosi, B., 2015a. Travel time model for shuttle-based storage and retrieval systems. Int. J. Adv. Manuf. Technol. 78, 1705–1725. https://doi.org/10.1007/s00170-014-6726-2

Lerher, T., Ekren, B.Y., Sari, Z., Rosi, B., 2015b. Simulation analysis of shuttle-based storage and retrieval systems. Int. J. Simul. Model. 14, 48–59. https://doi.org/10.2507/IJSIMM14(1)5.281

Lerher, T., Ficko, M., Palčič, I., 2021. Throughput performance analysis of Automated Vehicle Storage and Retrieval Systems with multiple-tier shuttle vehicles. Appl. Math. Model. 91, 1004–1022. https://doi.org/10.1016/j.apm.2020.10.032

Li, J., Huang, R., Dai, J.B., 2017. Joint optimisation of order batching and picker routing in the online retailer's warehouse in China. Int. J. Prod. Res. 55, 447–461. https://doi.org/10.1080/00207543.2016.1187313

Liu, Z., Wang, Y., Jin, M., Wu, H., Dong, W., 2020. Energy consumption model for shuttle-based Storage and Retrieval Systems. J. Clean. Prod. 282, 124480. https://doi.org/10.1016/j.jclepro.2020.124480

Man, X., Zheng, F., Chu, F., Liu, M., Xu, Y., 2019. Bi-objective optimization for a two-depot automated storage/retrieval system. Ann. Oper. Res. https://doi.org/10.1007/s10479-019-03222-1

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). Int. J. Prod. Res. 54, 4286–4304. https://doi.org/10.1080/00207543.2016.1144241

Manzini, R., Gamberi, M., Regattieri, A., 2006. Design and control of an AS/RS. Int. J. Adv. Manuf. Technol. 28, 766–774. https://doi.org/10.1007/s00170-004-2427-6

Marchet, G., Melacini, M., Perotti, S., 2015. Investigating order picking system adoption : a case-study-based approach 5567. https://doi.org/10.1080/13675567.2014.945400

Marchet, G., Melacini, M., Perotti, S., Tappia, E., 2013. Development of a framework for the design of autonomous vehicle storage and retrieval systems. Int. J. Prod. Res. 51, 4365–4387. https://doi.org/10.1080/00207543.2013.778430

MHI, 2021a. What is AS/RS. MHI. URL https://www.mhi.org/as-rs (accessed 5.26.21).

MHI, 2021b. Automated Storage and Retrieval Systems. MHI. URL https://www.mhi.org/fundamentals/automated-storage (accessed 5.26.21).

MHI, 2018. Strategic Solutions for Supply Chain. MHI. URL http://www.mhi.org/downloads/free/ACF6AE.pdf. (accessed 7.6.18).

Mital, P., Goetschalckx, M., Huang, E., 2015. Robust material handling system design with standard deviation, variance and downside risk as risk measures. Int. J. Prod. Econ. 170, 815–824. https://doi.org/10.1016/j.ijpe.2015.02.003

Noorul Haq, A., Karthikeyan, T., Dinesh, M., 2003. Scheduling decisions in FMS using a heuristic approach. Int. J. Adv. Manuf. Technol. 22, 374–379. https://doi.org/10.1007/s00170-002-1474-0

Pazour, J.A., Meller, R.D., 2014. A Frame Work and Analysis to Inform the Selection of Piece-level Order-fulfillment Technologies.

Poon, T.C., Choy, K.L., Chan, F.T.S., Ho, G.T.S., Gunasekaran, A., Lau, H.C.W., Chow, H.K.H., 2011. A real-time warehouse operations planning system for small batch replenishment

problems in production environment. Expert Syst. Appl. 38, 8524–8537. https://doi.org/10.1016/j.eswa.2011.01.053

Popović, D., Vidović, M., Bjelić, N., 2014. Application of genetic algorithms for sequencing of AS/RS with a triple-shuttle module in class-based storage. Flex. Serv. Manuf. J. 26, 432–453. https://doi.org/10.1007/s10696-012-9139-2

Ramtin, F., Pazour, J.A., 2015. Product allocation problem for an AS/RS with multiple in-the-aisle pick positions. IIE Trans. 47, 1379–1396. https://doi.org/10.1080/0740817X.2015.1027458

Rao, S.S., Adil, G.K., 2013. Optimal class boundaries, number of aisles, and pick list size for low-level order picking systems. IIE Trans. 45, 1309–1321. https://doi.org/10.1080/0740817X.2013.772691

Roodbergen, K.J., Vis, I.F.A., 2009. A survey of literature on automated storage and retrieval systems. Eur. J. Oper. Res. https://doi.org/10.1016/j.ejor.2008.01.038

Roodbergen, K.J., Vis, I.F.A., Taylor, G.D., 2015. Simultaneous determination of warehouse layout and control policies. Int. J. Prod. Res. 53, 3306–3326. https://doi.org/10.1080/00207543.2014.978029

Roshan, K., Shojaie, A., Javadi, M., 2019. Advanced allocation policy in class-based storage to improve AS/RS efficiency toward green manufacturing. Int. J. Environ. Sci. Technol. 16, 5695–5706. https://doi.org/10.1007/s13762-018-1921-6

Rouwenhorst, B., Reuter, B., Stockrahm, V., Van Houtum, G.J., Mantel, R.J., Zijm, W.H.M., 2000. Warehouse design and control: Framework and literature review. Eur. J. Oper. Res. 122, 515–533. https://doi.org/10.1016/S0377-2217(99)00020-X

Roy, D., Krishnamurthy, A., Heragu, S.S., Malmborg, C.J., 2017. A multi-tier linking approach to analyze performance of autonomous vehicle-based storage and retrieval systems. Comput. Oper. Res. 83, 173–188. https://doi.org/10.1016/j.cor.2017.02.012

Roy, D., Krishnamurthy, A., Heragu, S.S., Malmborg, C.J., 2015. Queuing models to analyze dwell-point and cross-aisle location in autonomous vehicle-based warehouse systems. Eur. J. Oper. Res. https://doi.org/10.1016/j.ejor.2014.09.040

Roy, D., Krishnamurthy, A., Heragu, S.S., Malmborg, C.J., 2012. Performance analysis and design trade-offs in warehouses with autonomous vehicle technology. IIE Trans. 44, 1045–1060. https://doi.org/10.1080/0740817X.2012.665201

Sari, Z., Saygin, C., Ghouali, N., 2005. Travel-time models for flow-rack automated storage and retrieval systems. Int. J. Adv. Manuf. Technol. 25, 979–987. https://doi.org/10.1007/s00170-003-1932-3

Schenone, M., Mangano, G., Grimaldi, S., Cagliano, A.C., 2020. An approach for computing AS/R systems travel times in a class-based storage configuration. Prod. Manuf. Res. 8, 273–290. https://doi.org/10.1080/21693277.2020.1781703

Silva, A., Coelho, L.C., Darvish, M., Renaud, J., 2020. Integrating storage location and order picking problems in warehouse planning. Transp. Res. Part E Logist. Transp. Rev. 140, 102003. https://doi.org/10.1016/j.tre.2020.102003

Sprock, T., Murrenhoff, A., McGinnis, L.F., 2017. A hierarchical approach to warehouse design. Int. J. Prod. Res. 55, 6331–6343. https://doi.org/10.1080/00207543.2016.1241447

Stadtler, H., 1996. An operational planning concept for deep lane storage systems. Prod. Oper. Manag. 5, 266–282. https://doi.org/10.1111/j.1937-5956.1996.tb00398.x

Statista, 2021. U.S. unit labor costs - quarterly change in the nonfarm business sector 2021. URL https://www.statista.com/statistics/217148/quarterly-percent-change-in-nonfarm-business-sector-unit-labor-costs-in-the-us/ (accessed 5.26.21).

Tappia, E., Roy, D., De Koster, R., Melacini, M., 2017. Modeling, analysis, and design insights for shuttle-based compact storage systems. Transp. Sci. 51, 269–295. https://doi.org/10.1287/trsc.2016.0699

Tappia, E., Roy, D., Melacini, M., De Koster, R., 2019. Integrated storage-order picking systems: Technology, performance models, and design insights. Eur. J. Oper. Res. 274, 947–965. https://doi.org/10.1016/j.ejor.2018.10.048

Tutam, M., White, J.A., 2019. A multi-dock, unit-load warehouse design. IISE Trans. 51, 232–247. https://doi.org/10.1080/24725854.2018.1488307

van den BERG, J.P., (NOUD) Gademann, A.J.R.M., 1999. Optimal routing in an automated

storage/retrieval system with dedicated storage. IIE Trans. 31, 407–415. https://doi.org/10.1080/07408179908969844

Vose, M.D., 1999. The simple genetic algorithm: foundations and theory. MIT Press.

Wang, Y., Mou, S., Wu, Y., 2015. Task scheduling for multi-tier shuttle warehousing systems. Int. J. Prod. Res. 53, 5884–5895. https://doi.org/10.1080/00207543.2015.1012604

Wen, U.P., Chang, D., Chen, S., 2001. The impact of acceleration/deceleration on travel-time models in class-based automated S/R systems. IIE Trans. 33, 599–608. https://doi.org/10.1080/07408170108936857

Wu, G., Xu, X., Gong, Y., De Koster, R., Zou, B., 2019. Optimal design and planning for compact automated parking systems. Eur. J. Oper. Res. 273, 948–967. https://doi.org/10.1016/j.ejor.2018.09.014

Wu, K.Y., Xu, S.S.D., Wu, T.C., 2013. Optimal scheduling for retrieval jobs in double-deep AS/RS by evolutionary algorithms. Abstr. Appl. Anal. 2013. https://doi.org/10.1155/2013/634812

Wu, Y., Zhou, C., Ma, W., Kong, X.T.R., 2020. Modelling and design for a shuttle-based storage and retrieval system. Int. J. Prod. Res. 0, 1–21. https://doi.org/10.1080/00207543.2019.1665202

Xu, X., Gong, Y., Fan, X., Shen, G., Zou, B., 2018. Travel-time model of dual-command cycles in a 3D compact AS/RS with lower mid-point I/O dwell point policy. Int. J. Prod. Res. 56, 1620–1641. https://doi.org/10.1080/00207543.2017.1361049

Xu, X., Shen, G., Yu, Y., Huang, W., 2015. Travel time analysis for the double-deep dual-shuttle AS/RS. Int. J. Prod. Res. 53, 757–773. https://doi.org/10.1080/00207543.2014.921351

Xu, X., Zhao, X., Zou, B., Gong, Y., Wang, H., 2019a. Travel time models for a three-dimensional compact AS/RS considering different I/O point policies. Int. J. Prod. Res. 0, 1–24. https://doi.org/10.1080/00207543.2019.1659519

Xu, X., Zhao, X., Zou, B., Li, M., 2019b. Optimal dimensions for multi-deep storage systems under class-based storage policies. Cluster Comput. 22, 861–875. https://doi.org/10.1007/s10586-018-2873-9

Yang, P., Miao, L., Xue, Z., Qin, L., 2015a. Optimal storage rack design for a multi-deep compact AS/RS considering the acceleration/deceleration of the storage and retrieval machine. Int. J. Prod. Res. 53, 929–943. https://doi.org/10.1080/00207543.2014.942441

Yang, P., Miao, L., Xue, Z., Ye, B., 2015b. Variable neighborhood search heuristic for storage location assignment and storage/retrieval scheduling under shared storage in multi-shuttle automated storage/retrieval systems. Transp. Res. Part E Logist. Transp. Rev. 79, 164–177. https://doi.org/10.1016/j.tre.2015.04.009

Yang, P., Yang, K., Qi, M., Miao, L., Ye, B., 2017. Designing the optimal multi-deep AS/RS storage rack under full turnover-based storage policy based on non-approximate speed model of S/R machine. Transp. Res. Part E Logist. Transp. Rev. 104, 113–130. https://doi.org/10.1016/j.tre.2017.05.010

Yu, Y., De Koster, R., 2012. Sequencing heuristics for storing and retrieving unit loads in 3D compact automated warehousing systems. IIE Trans. 44, 69–87. https://doi.org/10.1080/0740817X.2011.575441

Yu, Y., De Koster, R., 2009a. Designing an optimal turnover-based storage rack for a 3D compact automated storage and retrieval system. Int. J. Prod. Res. 47, 1551–1571. https://doi.org/10.1080/00207540701576346

Yu, Y., De Koster, R., 2009b. Optimal zone boundaries for two-class-based compact three-dimensional automated storage and retrieval systems. IIE Trans. 41, 194–208. https://doi.org/10.1080/07408170802375778

Yu, Y., De Koster, R., Guo, X., 2015. Class-based storage with a finite number of items: Using more classes is not always better. Prod. Oper. Manag. 24, 1235–1247. https://doi.org/10.1111/poms.12334

Zaerpour, N., De Koster, R., Yu, Y., 2013. Storage policies and optimal shape of a storage system. Int. J. Prod. Res. 51, 6891–6899. https://doi.org/10.1080/00207543.2013.774502

Zaerpour, N., Yu, Y., De Koster, R., 2017. Small is beautiful: A framework for evaluating and optimizing live-cube compact storage systems. Transp. Sci. 51, 34–51. https://doi.org/10.1287/trsc.2015.0586

Zaerpour, N., Yu, Y., De Koster, R., 2015. Storing fresh produce for fast retrieval in an

automated compact cross-dock system. Prod. Oper. Manag. 24, 1266–1284. https://doi.org/10.1111/poms.12321

Zammori, F., Neroni, M., Mezzogori, D., 2021. Cycle time calculation of shuttle-lift-crane automated storage and retrieval system. IISE Trans. 1–31. https://doi.org/10.1080/24725854.2020.1861391

Zhan, X., Xu, L., Ling, X., 2020. Task scheduling problem of double-deep multi-tier shuttle warehousing systems. Processes 9, 41. https://doi.org/10.3390/pr9010041

Zhao, Wang, Wang, Huang, 2019. Integer Programming Scheduling Model for Tier-to-Tier Shuttle-Based Storage and Retrieval Systems. Processes 7, 223. https://doi.org/10.3390/pr7040223

Zhao, X., Zhang, R., Zhang, N., Wang, Y., Jin, M., Mou, S., 2020. Analysis of the shuttle-based storage and retrieval system. IEEE Access 8, 146154–146165. https://doi.org/10.1109/ACCESS.2020.3014102

Zou, B., Gong, Y., Xu, X., Yuan, Z., 2017. Assignment rules in robotic mobile fulfilment systems for online retailers. Int. J. Prod. Res. 55, 6175–6192. https://doi.org/10.1080/00207543.2017.1331050

Zou, B., Xu, X., Gong, Y., De Koster, R., 2016. Modeling parallel movement of lifts and vehicles in tier-captive vehicle-based warehousing systems. Eur. J. Oper. Res. 254, 51–67. https://doi.org/10.1016/j.ejor.2016.03.039

APPENDICES

Appendix A. Comparison Between This Study and Yu and De Koster (2012)

Yu and De Koster (2012) studied the sequencing problem for a given batch of storage and retrieval tasks in a crane-based 3D AS/RS with conveyor-based DMM, in which each lane is equipped with a pair of conveyors for depth movement and an aisle-captive crane is used for moving SKUs between storage lanes and the I/O station, as shown in Figure S-1. The conveyors can pre-position the required storage location/retrieval tasks to the front end of the rack while the crane is moving/conducting other tasks. They considered both storage and retrieval tasks with the crane working in a DC mode (i.e., conducting one storage task and one retrieval task in one cycle). Since dummy storage/retrieval tasks can be added at the I/O station when the number of storage tasks and retrieval tasks are not equal, they only considered the scenario where the number of storage tasks equals to the number of retrieval tasks. A set of the open locations (S) and a set of locations of required SKUs (R) are given at the beginning. As tasks being processed, R gets smaller, and S changes as the location of the retrieved SKUs becomes open and the open location becomes unavailable after getting an SKU. Moreover, based on the proceeding DCs, the coordination of the open and retrieval locations also changes due to prepositioning. The optimal tasks schedule is to pair storage location and retrieval location for each DC. The problem is complex since the open and retrieval locations keep changing based on the preceding tasks. To track the impact of pre-positioning, infinite number of binary variables and strong nonlinear constraints were adopted by their model. To obtain a good solution in short time, multiple heuristics used for 2D systems were modified. However, all these heuristics preposition the open locations and retrieval locations with the same priority, and when there are more open locations than retrieval locations, preposition of retrieval tasks might be delayed. Therefore, the heuristic with the best performance was adapted by giving a higher priority for prepositioning retrieval location and the adapted heuristic outperformed the other heuristics significantly.

In our study, we use shuttles as DMMs and move shuttles across lanes if the number of shuttles is smaller than the number of lanes. We only consider retrieval tasks with a crane working in the SC mode. When a shuttle movement is required from lane i to lane j, the crane has to move from its dwell point to lane i, pick up the shuttle, move the shuttle to lane j, and wait at lane j for the SKU from there or move to another lane k for another retrieval/shuttle reallocation task. This process is

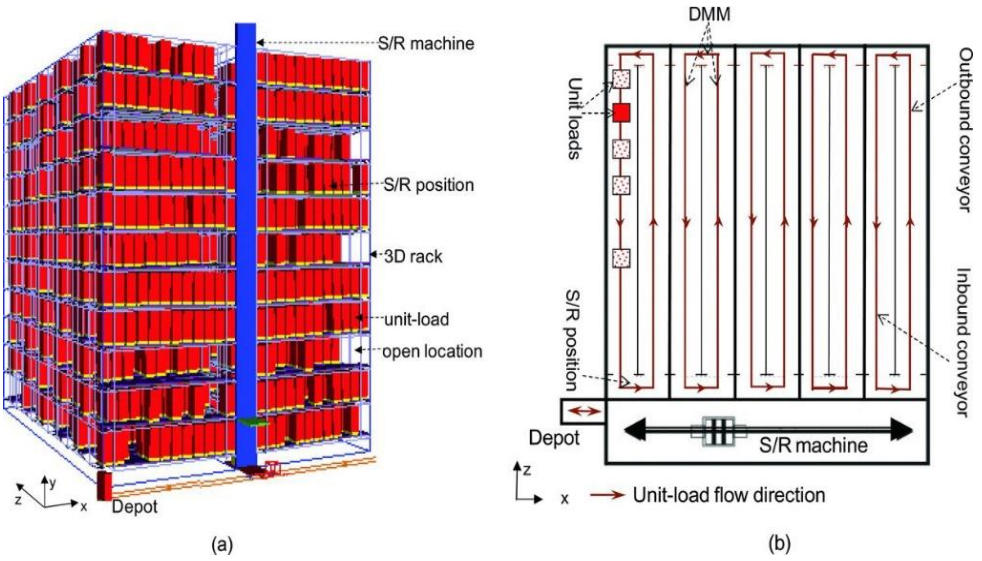


Figure S-1. An Overview of 3D AS/RS with Conveyor-based DMMs (Yu and De Koster, 2012)

equivalent to a DC in Yu and De Koster (2012) where the set of virtual open locations are $S = \{(i,j)|i \in N^{0'}\ \ j \in N^{1'}\}$. Here, $N^{0'}$ is the set of lanes with an available shuttle and $N^{1'}$ is the set of lanes needs a shuttle at the beginning of each DC cycle (i.e., shuttle reallocation). The time for conducting the virtual storage task (i.e., shuttle reallocation) will be $P_{o,i} + p_{ij}^2$. It is straightforward that the set of open locations, even though getting smaller, updates after each DC (i.e., shuttle reallocation). The position of retrieval locations will also change based on preceding tasks. By now, it looks like that our problem is mathematically equivalent to the one of Yu and De Koster (2012) after adding dummy storage (i.e., shuttle reallocation) tasks at the I/O station. However, there are additional restrictions in our study to add complexity. A shuttle can only be reallocated after all the retrieval tasks on that lane are finished. Therefore, where to insert the 'dummy storage' (i.e., shuttle reallocation) tasks matters in our study, and the open location cannot be selected arbitrarily. In other words, we must know when and where a shuttle will be available, which depends on the schedule of retrieval tasks. Therefore, our study is more complex than Yu and De Koster (2012). Furthermore, their model has infinite numbers of binary variables and strong nonlinear constraints that make the model almost impossible to be solved. In our study, an MIP with carefully defined variables and constraints was proposed to control the formulation size. Even still complicated, the proposed MIP is much simpler than their model and can be solved by commercial solvers for smaller cases.

Appendix B. MIP for Lane-to-Task Assignment Problem in 3D AS/RS

Only lanes with retrieval tasks are considered for the scheduling problem. However, in practice, a system may have more lanes than the SKU types and one type of goods is possibly stored in more than one lane. The lane-to-task assignment, which selects lanes to fulfill demand, can be modeled based on the MIP (2.1-2.21) proposed for the scheduling problem in this paper. In addition to the notations defined before, set A is defined as the set of SKU types, and N_a, $a \in A$, is the set of lanes used for storing SKU a. The demand on SKU type a is D_a. The lane-to-task assignment problem is formulated as follows based on the scheduling problem (2.1-2.21) by having parameter Q_i be a decision variable and adding an additional constraint set (B.22) to meet the demand.

Min (2.1)

s.t. $\sum_{i \in N_a} Q_i = D_a,$ $a \in A,$ (B.22)

(2.2-2.21)

$Q_i \geq 0.$

Appendix C. Compare with Model with Three-dimension Variables

Instead of y_{mi} and z_{mq}, a new type of binary variables a_{miq} can be created, which equals to one if the m^{th} SC task is for the q^{th} task from lane i; otherwise, $a_{miq} = 0$. The model with a_{miq} can be described as formulation (C.23-C.40), which has fewer constraints.

$$\text{Min } t_M + \sum_{i \in N} p_i^4 a_{MiQ_i} \tag{C.23}$$

s.t.

$$\sum_{i \in N/\{j\}} x_{ij} = 1, \qquad j \in N^1, \tag{C.24}$$

$$\sum_{j \in N^1} x_{ij} \le 1, \qquad i \in N, \tag{C.25}$$

$$\sum_{m=1}^{M} \sum_{q=1}^{Q'} a_{miq} = Q_i + \sum_{j \in N^1} x_{ij}, \qquad i \in N, \tag{C.26}$$

$$\sum_{i \in N} \sum_{q=1}^{Q'} a_{miq} = 1, \qquad m \in \{1, \dots, M\}, \tag{C.27}$$

$$u_i \ge m a_{miq}, \qquad i \in N, m \in \{1, \dots, M\}, q \in \{1, \dots, Q'\}, \tag{C.28}$$

$$l_i \le m a_{miq} + M\left(1 - a_{miq}\right), \qquad i \in N, m \in \{1, \dots, M\}, q \in \{1, \dots, Q'\}, \tag{C.29}$$

$$u_i \le l_j + M\left(1 - x_{ij}\right) \qquad i \in N, j \in N^1, \tag{C.30}$$

$$\sum_{m=1}^{M} \sum_{q=1}^{Q'} q a_{miq} = \sum_{q=1}^{Q_i+1} q + (Q_i + 1) \sum_{j \in N^1} x_{ij}, \qquad i \in N, \tag{C.31}$$

$$\sum_{k=1}^{M} a_{kiq} \le \sum_{m=1}^{M} a_{mi,q+1}, \qquad i \in N, q \in \{1, \dots, Q_i - 1\}, \tag{C.32}$$

$$\sum_{k=1}^{M} a_{kiQ_i} \le \sum_{m=1}^{M} a_{mi,Q_i+1} + M(1 - \sum_{j \in N^1} x_{ij}) \qquad i \in N, \tag{C.33}$$

$$t_1 \ge \sum_{i \in N^0} P_{i_0 i} a_{1i1}, \tag{C.34}$$

$$t_m \ge t_{m-1} + p_{ij}^3 \left(\sum_{q=1}^{Q_j+1} a_{mjq} + a_{m-1,i,Q_i} \right), \qquad i,j \in N, m \in \{2, \dots, M\}, \tag{C.35}$$

$$t_m \ge t_{m-1} + \sum_{k \in N} \left(p_{ik}^2 + p_{kj}^2\right)\left(x_{ik} + a_{m,i,Q_{i+1}} + \sum_{q=1}^{Q_j+1} a_{mjq} - 2\right), \qquad i,j \in N, m \in \{2, \dots, M\}, \tag{C.36}$$

$$r_{i1} \ge p_{i1}^1, \qquad i \in N^0, m \in \{1, \dots, M\}, \tag{C.37}$$

$$r_{i1} \ge t_m + p_{ij}^2 + p_{i1}^1 + T\left(a_{mj,Q_j+1} - x_{ji} - 2\right), \qquad i \in N^1,\ j \in N, m \in \{1, \dots, M\}, \tag{C.38}$$

$$r_{iq} \ge t_m + p_{iq}^1 - T\left(1 - a_{mi,q-1}\right), \qquad i \in N, q \in \{2, \dots, Q_i + 1\}, m \in \{1, \dots, M\}, \tag{C.39}$$

$$t_m \ge r_{iq} - T\left(1 - a_{miq}\right), \qquad i \in N, q \in \{1, \dots, Q_i + 1\}, m \in \{1, \dots, M\}, \tag{C.40}$$

$$t_m, r_{iq}, u_i, l_i \ge 0; a_{miq}, x_{ij} \in \{0,1\}. \tag{C.41}$$

The objective function (C.23) minimizes the total cycle time for the crane to complete a given batch of retrieval tasks. Constraint set (C.24) makes sure that a lane in N^1 receives a shuttle exactly once. Constraint set (C.25) guarantees that the crane moves a shuttle out of a lane in N^0 up to once. Constraint set (C.26) assures that the number of SC tasks related to lane i is the number of retrieval tasks at lane i plus one shuttle reallocation task if there is any. Constraint set (C.27) ensures that the crane only performs one retrieval or shuttle reallocation task during one SC. Constraint sets (C.28) through (C.30) make sure that the crane can only serve a lane that has a shuttle and avoid the sub-tour problem. In other words, when a shuttle is moved from lane i to

lane j, all tasks in lane i should be finished before the first task in lane j starts. Constraint sets (C.31) through (C.33) make sure that the SKUs are retrieved following first-in-last-out rule. Constraint sets (C.34) through (C.36) make sure that the m^{th} SC task cannot start until the $(m-1)^{\text{th}}$ SC task is completed. Specifically, constraint set (2.34) is for the case of the first SC task; (C.35) is for the case when the $(m-1)^{\text{th}}$ SC task is to retrieve an SKU; and (C.36) is for the case when the $(m-1)^{\text{th}}$ SC task is to move a shuttle to lane j. Constraint sets (C.37) through (C.39) are used to obtain the time when the shuttle with the SKU for the q^{th} task at lane i is ready for pick up by the crane. Constraint set (C.37) is used for the first retrieval task on lanes in N^0; (C.38) is used for the first retrieval task on lanes from N^1; and (C.39) makes sure that the shuttle cannot move for the q^{th} SKU before the crane picks up the $(q-1)^{\text{th}}$ SKU from that lane. Constraint set (C.40) calculates the moment when the crane picks up the SKU or shuttle in its m^{th} cycle.

Let's define the MIP model represented by formulation (2.1-2.21) as MIP_1 and the MIP (C.23-C.40) with three dimensional variables as MIP_2. Numerical experiments were conducted to compare the computational performance of these two models. Due to the computational burden for solving large case problems, only small cases were tested. Gurobi was used to solve both formulations with a running time limit of 600 seconds. When it reaches the time limitation, Gurobi stopped and returned a feasible solution if it could find one. The comparison is demonstrated by Table S-1, where $T_{MIP_i}, VN_{MIP_i}, S_{MIP_i}$, and G_{MIP_i} represent the objective value, number of binary variables, solving time and gap to lower bound found by Gurobi of $MIP_i \in \{MIP_1, MIP_2\}$, respectively.

As demonstrated by Table S-1, MIP_2 has much more binary variables than MIP_1. The difference increase dramatically as the problem size gets larger, which significantly increased the computational burden. When the problem size is small enough, both MIP_1 and MIP_2 can be solved to optimum in a short time. However, as the problem size getting larger, MIP_1 may be solved to a better feasible solution (e.g., $(\sum_{i\in N} Q_i, |N|, |N^0|) = (38, 13, 8)$). Even though sometimes Gurobi found similar feasible solution for these two models (e.g., $(\sum_{i\in N} Q_i, |N|, |N^0|) = (15, 10, 3)$), MIP_1 has a much smaller gap to its lower bound. For five cases, Gurobi cannot obtain any feasible solution for MIP_2 within 600 seconds.

Table S-1. Comparison between MIP_1 and MIP_2

$(\sum_{i\in N} Q_i, \lvert N \rvert, \lvert N^0 \rvert)$	T_{MIP_1}	T_{MIP_2}	VN_{MIP_1}	VN_{MIP_2}	S_{MIP_1}	S_{MIP_2}	G_{MIP_1}	G_{MIP_2}
(3,3,1)	29	29	25	30	0.1	0.1	0%	0%
(10,8,1)	219	221	187	408	600	600	53%	98%
$(74, 11, 6)$	894	N/A	1738	9559	600	600	70%	N/A
(83,11,9)	1059	N/A	2125	13090	600	600	72%	N/A
$(103, 17, 4)$	N/A	N/A	3480	25636	600	600	N/A	N/A
$(38, 13, 8)$	457	468	817	3354	600	600	69%	99%
$(36, 8, 7)$	268	268	703	3256	600	600	52%	88%
$(100, 18, 3)$	2396	N/A	3680	28980	600	600	84%	N/A
$(15, 10, 3)$	184	184	308	880	600	600	60%	99%
(32, 19, 3)	801	N/A	1104	3648	600	600	80%	N/A

Appendix D. Proof of Theorem 2.1 and Theorem 2.2

Theorem 2.1. *The 3D AS/RS retrieval task scheduling problem is NP-hard.*

Proof: According to Han et al. (1987), the task scheduling problem in 2D AS/RS operating in the DC mode with multiple open locations is equivalent to the traveling salesman problem (TSP) in its simplified version in which there is only one open location. Since the TSP is NP-hard (Cormen et al. 2009), the task scheduling problem in 2D AS/RS is NP-hard. For the 3D AS/RS scheduling problem considered in our study, if the shuttle speed is assumed to be infinite, the crane can pick up an SKU/shuttle instantly when arrives at a target lane. In that case, after moving a shuttle to a lane, the crane will pick up an SKU from the lane immediately instead of going to serve another lane. If we consider all locations of shuttles as open locations, the crane operates in the DC mode when transferring shuttles: starts from the I/O point to an open location, picks up the shuttle there and transfers it to lane j, hands over the shuttle and picks up an SKU form lane j, and travels back to I/O point. It is obvious that the problem considered by Han et al. (1987) is a special case of our problem with infinite shuttle speed. As this special case is NP-hard, our scheduling problem for the 3D AS/RS is also NP-hard. □

Theorem 2.2: Optimality Condition. *Consider the m^{th} and $(m+1)^{th}$ tasks in an optimal task schedule. If the m^{th} task is for retrieving the $(q_j)^{th}$ SKU from lane j, the $(m+1)^{th}$ task is for retrieving the $(q_k)^{th}$ SKU from lane k, and the $(m-1)^{th}$ task is associated with lane i, we should always have $p_m \geq s_m$ where $p_m = max\{r_{k,q_k} - t_{m-1} - p^3_{ik}, 0\}$ and $s_m = max\left\{r_{j,q_j} - t_{m-1} - p^3_{ij}, 0\right\}$ if the $(m-1)^{th}$ task is a retrieval task, otherwise; $p_m = max\{r_{k,q_k} - t_{m-1} - p^2_{i,l} - p^2_{l,k}, 0\}$ and $s_m = max\left\{r_{j,q_j} - t_{m-1} - p^2_{i,l} - p^2_{l,j}, 0\right\}$ if the $(m-1)^{th}$ task is to reallocate a shuttle from lane i to l.*

Proof: Suppose the $(m-1)^{th}$ task is for retrieving an SKU, $p_m = \max\{r_{k,q_k} - t_{m-1} - p^3_{ik}, 0\}$ and $s_m = \max\left\{r_{j,q_j} - t_{m-1} - p^3_{ij}, 0\right\}$. In addition, Let's define w_m and w_{m+1} as the crane's waiting time of the m^{th} and $m+1^{th}$ task. Clearly, $w_m = s_m$ and $w_{m+1} = \max\{r_{k,q_k} - t_m - p^3_{jk}, 0\}$. To prove the optimality, we have to show that the total waiting time of these two tasks will increase if we switch the sequence. Let's define w'_m and w'_{m+1} as the crane's waiting time of the new m^{th} and $m+1^{th}$ task after switching the sequence. It is clear that when $s_m = w_m = 0$, the

solution is optimal, since we can always have $w_m = w'_{m+1} = 0$ and $w'_m \geq w_{m+1}$. When $s_m > 0$, two cases can be considered.

Case 1: $p_m = 0, r_{k,q_k} \leq t_{m-1} + p^3_{ik}$

In that case, we can have

$$w'_m = \max\{p_m, 0\} = 0,$$

and

$$w'_m + w'_{m+1} = w'_{m+1} = \max\left\{r_{j,q_j} - t_{m-1} - p^3_{ik} - p^3_{jk}, 0\right\} < r_{j,q_j} - t_{m-1} - p^3_{ij}.$$

Clearly, when $s_m > 0$, we must have $p_m > 0$ to guarantee the optimality.

Case 2: $p_m > 0$

When $p_m > 0$, $w'_m = r_{k,q_k} - t_{m-1} - p^3_{ik} > 0$, and $w'_{m+1} = \max\left\{r_{j,q_j} - r_{k,q_k} - p^3_{jk}, 0\right\}$. If $w'_{m+1} > 0$, which is equivalent to $r_{k,q_k} - r_{j,q_j} < -p^3_{jk}$, $w_{m+1} = \max\left\{r_{k,q_k} - r_{j,q_j} - p^3_{jk}, 0\right\} = 0$. Therefore, after switching, the total waiting time of these two tasks will be

$$w'_m + w'_{m+1} = r_{j,q_j} - t_{m-1} - p^3_{ik} - p^3_{jk} < r_{j,q_j} - t_{m-1} - p^3_{ij} = w_m + w_{m+1} = w_m.$$

To guarantee the optimality of the given solution, $r_{k,q_k} - r_{j,q_j} \geq -p^3_{jk}$ and $w'_{m+1} = 0$. In that case, if $w_{m+1} = 0$ and $r_{k,q_k} - r_{j,q_j} \leq p^3_{jk}$,

$$w'_m + w'_{m+1} = w'_m = r_{k,q_k} - t_{m-1} - p^3_{ik}, \text{ and}$$

$$w_m + w_{m+1} = w_m = r_{j,q_j} - t_{m-1} - p^3_{ij}.$$

To guarantee the optimally,

$$r_{k,q_k} - t_{m-1} - p^3_{ik} \geq r_{j,q_j} - t_{m-1} - p^3_{ij},$$

which is equivalent to

$$p^3_{jk} \geq r_{k,q_k} - r_{j,q_j} \geq p^3_{ik} - p^3_{ij}.$$

And if $w_{m+1} > 0$,

$$w'_m + w'_{m+1} = w'_m = r_{k,q_k} - t_{m-1} - p^3_{ik} > r_{k,q_k} - t_{m-1} - p^3_{ij} - p^3_{jk} = w_m + w_{m+1}.$$

According to the discussion, if $s_m = 0$, p_m can take any value. However, when $s_m > 0$, we need $r_{k,q_k} - r_{j,q_j} \geq p_{ik}^3 - p_{ij}^3$, which is equivalent to $p_m \geq s_m$. The same logic can be applied for the scenario when $m - 1^{th}$ task is for reallocating a shuttle.

Appendix E. Example of the Genetic Algorithm

Consider a system with 6 lanes, where lane 1, 3 and 4 belong to N^0 and the others do not have a shuttle at the beginning. The retrieval task number from each lane can be found in Table S-2. Therefore, there are totally 13 retrieval tasks and three shuttle reallocation tasks.

To generate the initial feasible solutions (chromosomes), short-form chromosomes are created first to represent the shuttle assignment. In a short-form chromosome, each gene corresponds to a lane with the order. All shuttles are numbered with a unique integer number $i \in \{1, \dots, |N^0|\}$. The gene of each lane in N^0 (i.e., each lane with a shuttle at the beginning) is the index of the shuttle at that lane. In the example shown in Figure S-2, lane 1 has shuttle 1 at the beginning and therefore has the value of 1 in the short-form chromosome, while shuttle 2 and shuttle 3 are in lane 3 and lane 4 at the beginning, respectively. The gene of each lane in $|N^1|$ is comprised by two parts, the integer and fractional parts. The integer part, ranging from 1 to $|N^0|$, represents which shuttle is assigned to that lane. The fractional part is originally randomly generated in range $(0, 1)$ without any repeats and represents the sequence of shuttle movements. Shuttles are moved across lanes following an ascending order of the fractional part. In the example short-form chromosome of Figure S-2, shuttle 1 is moved by the crane from lane 1 to lane 2 and then to lane 6. Shuttle 2 serves lane 3 and then lane 5. Shuttle 3 stays at lane 4.

In the long-form chromosome, each gene still represents a lane but will be assigned a list of numbers. For each number assigned to a lane, the integer part equals to the integer part of the number assigned to this lane/gene in short-from chromosome. Fractional part of each number is originally generated randomly in the range of 1×10^{-D} to $M \times 10^{-D}$, where M is the number of total SC tasks and D is $\lceil \log_{10} M \rceil$. Please see Figure S-2 for an example of a long-form chromosome. The shuttle assignment is as the information in the short-from chromosome, while the size of the fractional part expresses the sequence the task served by the crane. For instance, the sequence (1.02, 1.04, 1.05) for lane 1 means that shuttle 1 is used for lane 1, and these three tasks are scheduled as the second, fourth and fifth crane's task, respectively. Given the fact that lane 1 only has two required SKUs, the fifth crane's cycle is for reallocating the shuttle at lane 1.

Table S-3 shows the corresponding order of SC tasks across lanes (i.e., crane's route). The coding of a long-form chromosome represents both the shuttle assignment and SC task sequencing. For

Table S-2. Details of Examples

Lane i	Retrieval Tasks
1	2
2	3
3	1
4	2
5	3
6	2

Short-form Chromosome:

$[1, 1.28, 2, 3, 2.45, 1.94]$

Long-form Chromosome:

$\big[[1.02, 1.04, 1.05], [1.11, 1.12, 1.13, 1.14], [2.03, 2.06], [3.01, 3.1], [2.07, 2.08, 2.09], [1.15, 1.16]\big]$

Figure S-2. Example of Generating an Initial Chromosome

Table S-3. SC Task Schedule of the Example in Figure S-2

SC Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Lane	4	1	3	1	1	3	5	5	5	4	2	2	2	2	6	6

the following steps, we can always find the short-from chromosome from a long-form chromosome by extracting the first element of each list in the offspring and setting the number assigned to lane i in N^0 to be the number assigned to the shuttle on lane i.

Figure S-3 illustrates a crossover operation in which the genes for lanes 2 and 4 are both swapped between two parent chromosomes. An offspring (temporary) generated by a crossover may be infeasible so that modifications are needed. For instance, Temporary Offspring 2 has the crane start to serve lane 5 before the lane receives its assigned shuttle from lane 4. Even the total number of tasks may not equal M. This is the case for both Temporary Offspring 1 and 2 in Figure S-3. Therefore, a modification might be conducted after a crossover. Each temporary offspring is first checked to see if the corresponding short-form chromosome is illegal.

An extracted short-form chromosome might be illegal if a shuttle has to serve two lanes simultaneously, which corresponds to the situation that the same number is assigned to multiple lanes in N^1. Considering the short-form chromosome in Figure S-4, lane 2 and 6 have the same value assigned but shuttle 1 cannot serve these two lanes simultaneously. An infeasible short-form chromosome can be corrected by assigning a new number while keeping the same integer part but a different fractional part. The new assigned number should not change the relative sequence of lanes served by the same shuttle. For the illegal short-from chromosome of $[1, 1.11, 2, 3, 1.07, 1.11]$, as an example, new fractional parts should be assigned to lane 2 and 6 after serving lane 5. However, after correcting it, lane 2 and 6 should not get a shuttle earlier than lane 5. In another word, the new value assigned to lane 2 and 6 must be larger than 1.07 and less than 2.0.

After checking and correcting the short-form chromosome, we check the temporary offspring to see whether any of the following situations occur: 1) Multiple elements in an offspring have the same fractional part; 2) The lane that is served earlier than other lanes by the same shuttle has elements that are equal to or greater than some elements from the genes corresponding to these lanes; and 3) The number of elements assigned to one lane does not equal to the number of task on that lane. The elements causing any of these situations will get new fractional parts, offspring

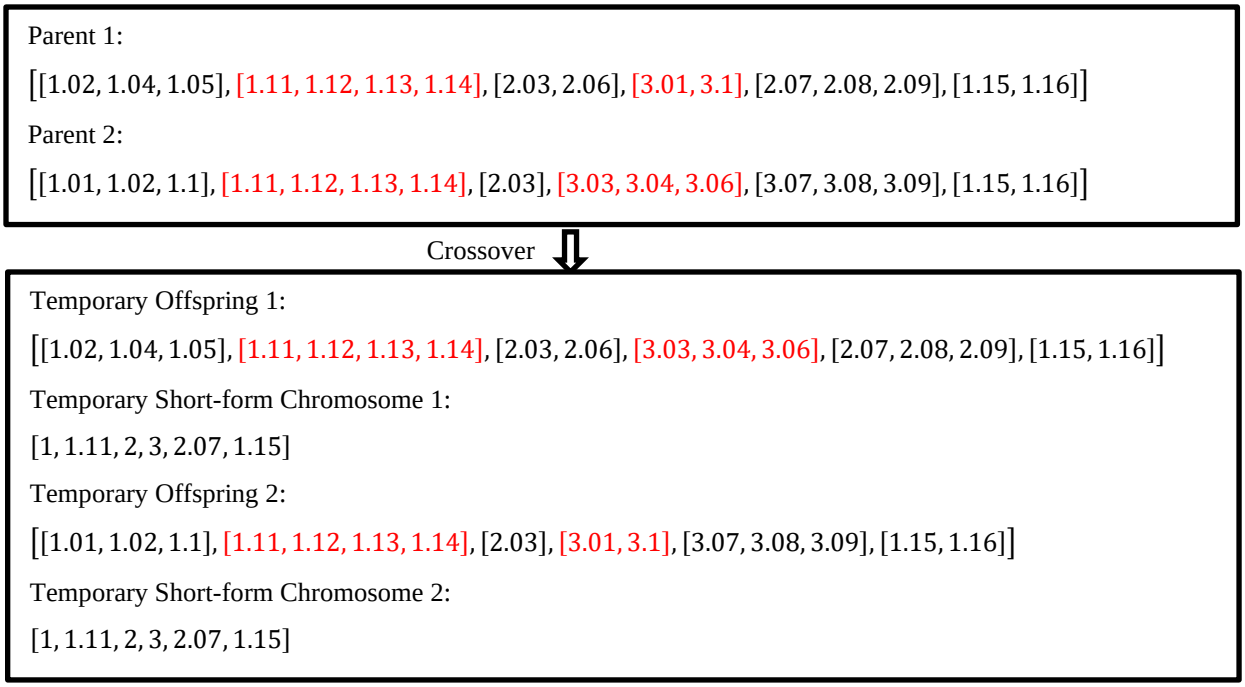


Figure S-3. Example of Crossover

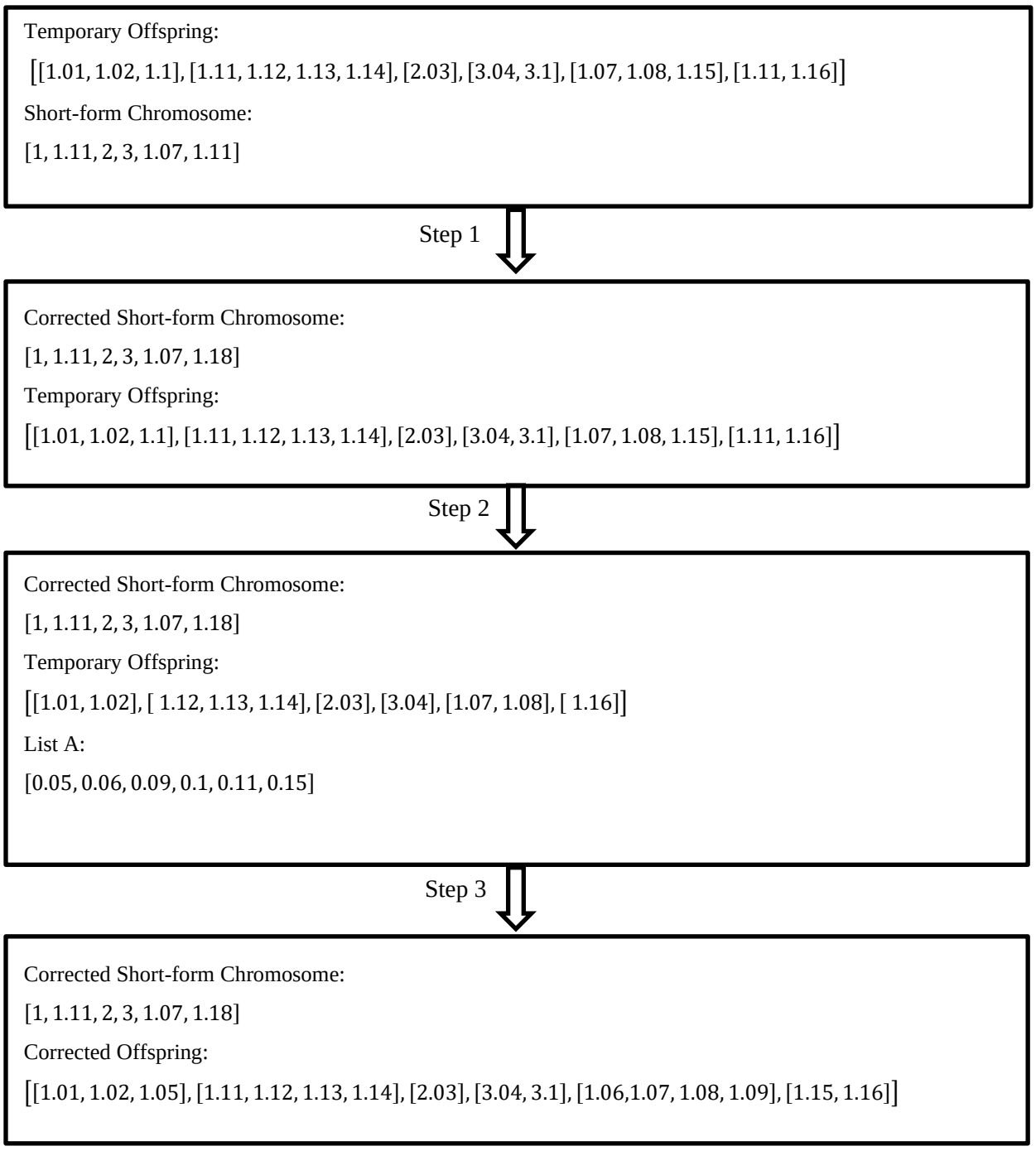


Figure S-4. Example of Checking and Modifying an Offspring

should be consistent with the information carried by the corrected short-form chromosome. The algorithm of correcting an illegal chromosome is shown below

Step 1: Extract the temporary short-short chromosome from the temporary offspring;
    While the same number is assigned to multiple lanes in N^1 in the temporary short-from chromosome:
        Randomly generate fractional part in range $(0, 1)$;
    While assigning the new generated factional number change relative sequence of lanes served by the same shuttle:
        Randomly generate fractional part in range $(0, 1)$;
        Replace the fractional parts of illegal genes with the new generated fractional part;
    Return the corrected short-from chromosome and go to step 2;
Step 2: Create an empty list A,
  While the same fractional parts are assigned to different elements from the temporary offspring:
        Remove these elements;
    While a gene/lane is assigned more elements/numbers than the number of tasks on it:
            Randomly remove elements from that gene until the assigned numbers equal to the number of tasks on it;
        While a lane that is severed earlier than other lanes by the same shuttle has elements that are larger than some elements from lists corresponded to those lanes:
            Remove the elements that cause the problem from genes;
    If any fraction part in range of 1×10^{-D} to $M \times 10^{-D}$ is missed in the temporary offspring:
        Add the missed fractional parts to list A;
    Go to step 3
Step 3: While list A is not empty:
        Try Assign the fractional parts from list A to genes which need one;
        If the corrected offspring becomes feasible:
            Empty list A;
        Return the corrected offspring and terminate.

Figure S-4 shows how the algorithms work according to the corrected short-form chromosome: shuttle 1 will be moved from lane 1 to lane 5, lane 2 and then lane 6. Shuttle 2 and 3 stays at lane 3 and lane 4 respectively. There are 4 errors in the temporary offspring: 1) Tasks from lane 2 and lane 6 are all marked as the 11^{th} task of the crane; and tasks from lane 1 and lane 4 are all marked as the 10^{th} task of the crane; 2) There should be 3 retrieval tasks plus one shuttle movement task on lane 5, however, only 3 elements are assigned to this lane; and 3) The crane starts to take care of tasks from lane 5 and lane 2 before these lanes getting the assigned shuttle. The elements cause those problems will be removed from the temporary offspring. Based on the

way we create a chromosome, we know that all the fractional part in the range 1×10^{-D} to $M \times 10^{-D}$ should appear exactly once in the offspring. The missed fractional parts will be added in to list A. Considering the example, list A should be $[0.05, 0.06, 0.1, 0.11, 0.15]$. Then we will try to assign the fractional parts to the genes which need one until we get a feasible offspring that is also consistent with the shuttles assignment decided by the corrected short-form chromosome.

The GA mutation operator is applied on short-form chromosomes extracted from the long-form chromosomes. One lane $i \in N^1$ (e.g., lane 2) will be randomly selected and its corresponded gene from the short-form chromosome will get a new number (shuttle assignment), as shown in Figure S-5. The new assignment number is generated as the way we did in chromosome representation. The new generated number should not be equal to numbers already assigned to other genes. As in the crossover, a modification is needed as the associated long-form chromosome may be infeasible after a mutation. Therefore, with the same method used for checking offspring, the long-form chromosome is checked and revised according to the information carried by the mutated short-form chromosome.

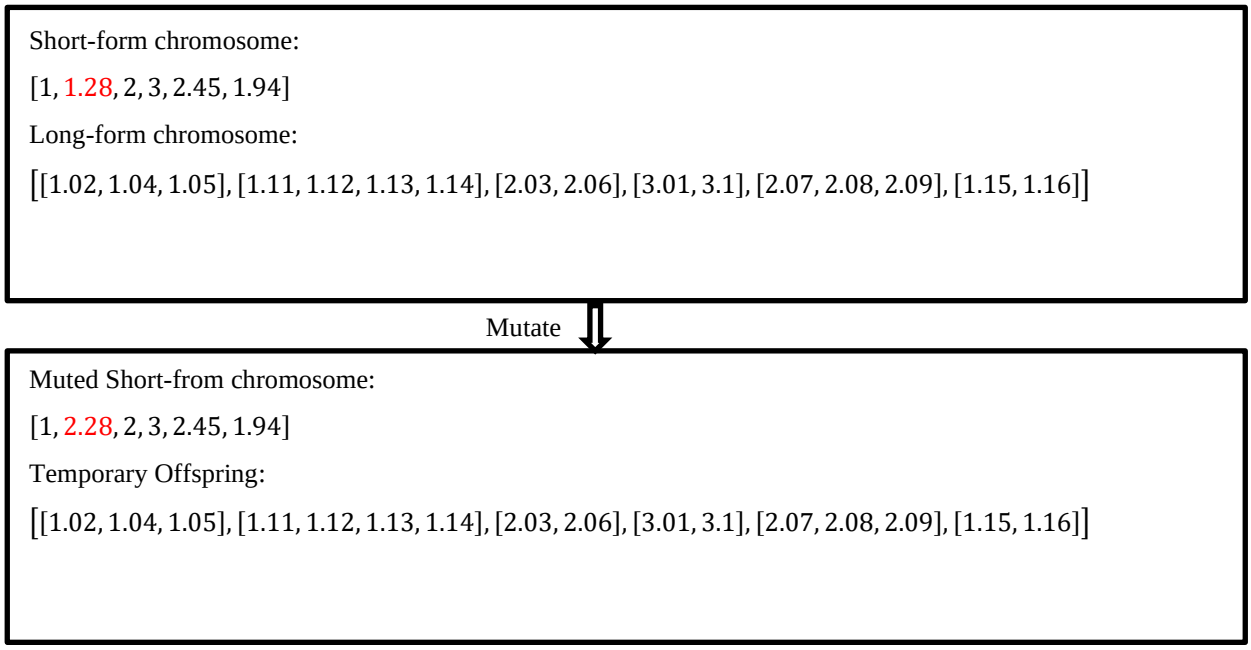


Figure S-5. Example of Mutation

Appendix F. Proof to Theorem 3.3

Proof. The theorem can be proven by looking at the difference $ESC_{\mathcal{R}_1} - ESC_{\mathcal{R}_2}$, especially the detailed time components defined by equations (3.12-3.14): expected time for the lift to travel between tiers for retrieving an SKU ($TSC_r^{\mathcal{R}}$), expected time for moving a shuttle ($TSC_m^{\mathcal{R}}$), and expected time waiting for the shuttle to move the SKU ($TSC_w^{\mathcal{R}}$), where $\mathcal{R} \in \{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3\}$ represents the shuttle dispatching rule. Among three, $TSC_r^{\mathcal{R}}$ is constant once the storage policy is fixed. We can write the difference in shuttle movement time, $TSC_m^{\mathcal{R}_1} - TSC_m^{\mathcal{R}_2}$, as

$$\Delta(M) = \frac{N-M}{N^2(N-1)} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} c_{ij} - t_{oi} - \frac{1}{N(N-M+1)} \sum_{i=1}^{N-M+1} \sum_{\substack{j=1 \\ j \neq i}}^{N-M+1} c_{ij} - t_{oi}, \tag{F.29}$$

which is a function of M. It is clear that $\Delta(1) = \Delta(N) = 0$. For $1 < M < N$, we can write $d(M) = \Delta(M+1) - \Delta(M)$ as

$$d(M) = \frac{1}{N(N-M+1)} \sum_{i=1}^{N-M+1} \left(\sum_{\substack{j=1 \\ j \neq i}}^{N-M+1} c_{ij} - t_{oi} \right) - \frac{1}{N(N-M)} \sum_{i=1}^{N-M} \left(\sum_{\substack{j=1 \\ j \neq i}}^{N-M} c_{ij} - t_{oi} \right) - \frac{1}{N^2(N-1)} \left(\sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} c_{ij} - t_{oi} \right).$$

Then, we can have

$$d(M+1) - d(M)$$

$$= \frac{2}{N(N-M)} \sum_{i=1}^{N-M} \sum_{\substack{j=1 \\ j \neq i}}^{N-M} t_{ij} - \frac{1}{N(N-M-1)} \sum_{i=1}^{N-M-1} \sum_{\substack{j=1 \\ j \neq i}}^{N-M-1} t_{ij} - \frac{1}{N(N-M+1)} \sum_{i=1}^{N-M+1} \sum_{\substack{j=1 \\ j \neq i}}^{N-M+1} t_{ij}.$$

Let $k = N - M$, we can have

$$N\big(d(M+1) - d(M)\big) = \frac{4(k^2-1)\left(\sum_{i=1}^{k-1}\sum_{j=i+1}^{k} t_{ij}\right) - 2(k^2+k)\left(\sum_{i=1}^{k-2}\sum_{j=i+1}^{k-1} t_{ij}\right) - 2(k^2-k)\left(\sum_{i=1}^{k}\sum_{j=i+1}^{k+1} t_{ij}\right)}{k(k+1)(k-1)} =$$

$$\frac{2(k^2-k-2)\left(\sum_{i=1}^{k-1} t_{ik}\right) - 2(k^2-k)\left(\sum_{i=1}^{k} t_{ik}\right)}{k(k+1)(k-1)} < 0,$$

which means $\Delta(M)$ is concave, and because $\Delta(1) = \Delta(N) = 0$, $\Delta(M) = TSC_m^{\mathcal{R}_1} - TSC_m^{\mathcal{R}_2} \geq 0$. The same logic can be applied to show the lift waiting time difference, $TSC_w^{\mathcal{R}_1} - TSC_w^{\mathcal{R}_2}$, is

concave in terms of M. Here, we only show $TSC_w^{\mathcal{R}_1} - TSC_w^{\mathcal{R}_2} \geq 0$. We can write $TSC_w^{\mathcal{R}_1} - TSC_w^{\mathcal{R}_2}$ as

$$TSC_w^{\mathcal{R}_1} - TSC_w^{\mathcal{R}_2} = \sum_{i=1}^{N-M+1} \frac{(M-1)(N-M)[t^r - t_{oi}]^+}{N^2(N-M+1)} - \sum_{i=N-M+2}^{N} \frac{N-M}{N^2}[t^r - t_{oi}]^+ \quad \text{(F. 30)}$$

Please note that $[t^r - t_{o1}]^+ \geq [t^r - t_{o2}]^+ \geq \cdots \geq [t^r - t_{oN}]^+$. By defining $E = \frac{\sum_{i=N-M+2}^{N}[t^r - t_{oi}]^+}{M-1}$, we have

$$TSC_w^{\mathcal{R}_1} - TSC_w^{\mathcal{R}_2} \geq \frac{(M-1)(N-M)}{N^2}E - \frac{(M-1)(N-M)}{N^2}E = 0$$

and the equality only holds when $M = 1$ or N or $t^r \leq t_{o1}$. Therefore, $ESC_{\mathcal{R}_1} - ESC_{\mathcal{R}_2} \geq 0$ under random storage and the difference is concave in terms of M. ■

Appendix G. Comparison of Shuttle Dispatching Rules under DC Operations with Slower Lift

As discussed in the main text, $\mathcal{R}_1$ can be dominated by $\mathcal{R}_2$ if the system has slower lift. The statement id demonstrated by Figure S-6, where the speed profile of the lift becomes $v_l = 0.5\ m/s$, and $a_l = 0.5\ m/s^2$. However, when the demand rate is heterogeneous (i.e., $G(0.1)$ is high), $\mathcal{R}_3$ is still the best.

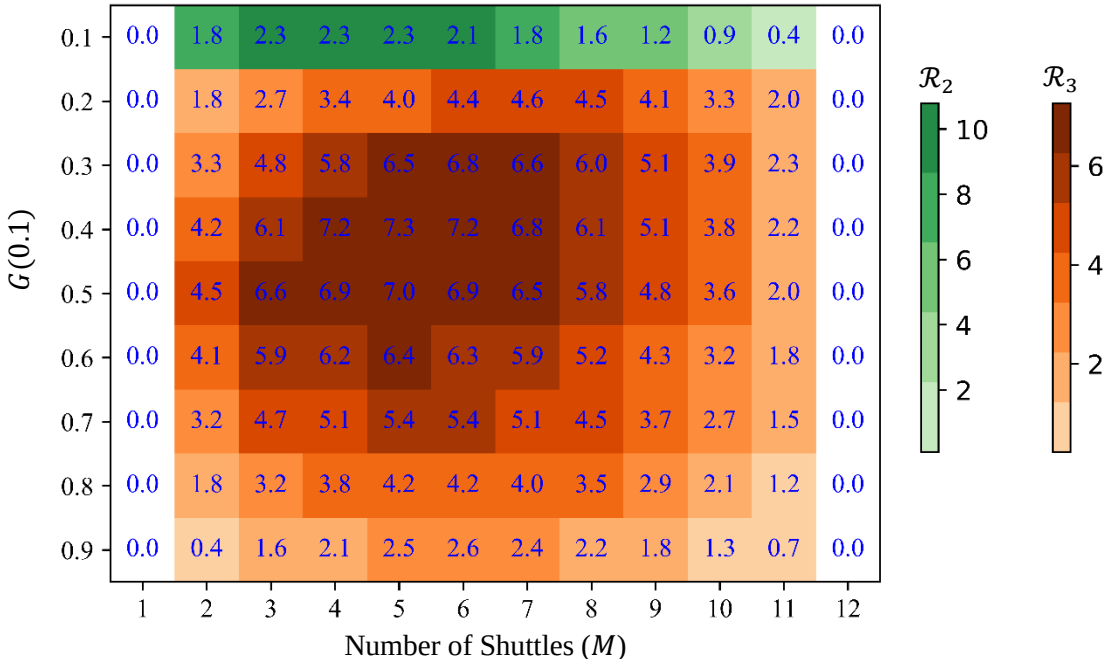


Figure S-6. Best Dispatching Rules under Various $G(0.1)$ and M Values under Class-based Storage, DC Operations, and Slower Lift

Appendix H. Estimating Cost Parameters

The prices of cranes and shuttles are considered as $\$200{,}000/unit$ and $\$800/unit$ (Alibaba.com, 2021a, 2021b). By assuming five years lifetime for shuttles and cranes, the daily equipment cost is $\$109/unit/day$ and $\$0.4/unit/day$. Considering the daily operation and maintenance cost, and the land cost for aisles, the cost of having one crane (C_v) and one shuttle (C_s) are estimated to $\$130/unit/day$ and $\$0.6/unit/day$. The daily warehouse cost (e.g., land cost, rack cost, energy cost) is assumed as $\$0.1/m^3/day$. Even though we did not give an exact value of C_t since it varies for different business scenarios, we describe the method of estimating C_t.

Assume the warehouse is expected to finish 2,000 retrieving tasks per day and the warehouse operates eight hours a day, the excepted cycle time (benchmark) to meet the desired throughput capacity should be $8 \times \frac{3600}{2000} = 14.4$ seconds. Assume the cost/penalty for not meeting the desired throughput capacity by one SKU is $\$0.01$, and the warehouse has an average cycle time of 15.4 seconds, then the warehouse's real throughput capacity is 1,870 retrieval tasks per day. Therefore, the cost of having one more second than the benchmark expected cycle time is $0.01 \times (2{,}000 - 1{,}870) = 1.3\$/day$.

Appendix I. Example of Branch-and-Bound Tree

Consider a warehouse design problem with two types of SKUs (i.e., $I = \{1,2\}$), and $Q_1 = 4$ and $Q_2 = 3$. The number of storage lanes on each rack is considered as two (i.e., $n = 2$), and the system allows a maximum of three racks. The branch-and-bound tree following the standard branch-and-bound procedure is shown in Figure S-7. However, the branch-and-bound tree under the standard B&B algorithm is too large (i.e., 34 nodes), and we only show the branches start from the node [2]. Please note that even though some nodes are representing extremely unreasonable warehouse design, these branches can be pruned quickly by comparing lb and UB.

Compared with the standard B&B algorithm, the modified B&B algorithm can result in a much smaller tree. By solving $D(n_r)$ we have $D(1) = 4, D(2) = 4$, and $D(3) = 4$. Following the modified B&B algorithm, the branch-and-bound tree is shown in Figure S-8. Also, the difference in the sizes of the trees under these two B&B algorithms will be much significant as the increase of the number of allowed racks and inventory levels.

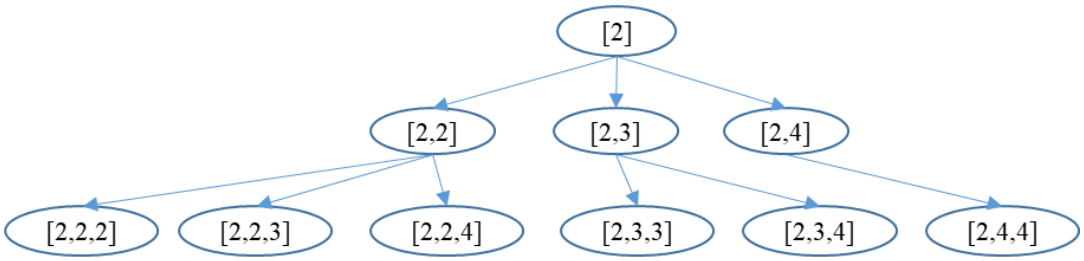


Figure S-7. Part of the B&B Tree under the Standard B&B Algorithm

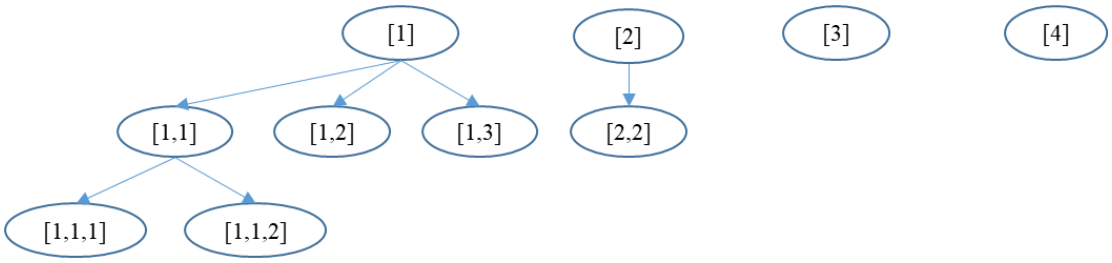


Figure S-8. B&B Tree under the Modified B&B Algorithm

Appendix J. MIP for SKU-to-Racks Assignment Problem with Given Racks Configuration

Assume a given racks configuration $[k_1, k_2, .., k_{n_r}]$ with n_r racks, the SKU-to-rack assignment problem can be formulated as an MIP defined by Equation (J.34-J.43), which is the same as the MIP defined by Equation (4.6) and (4.19-4.30) except the number of racks and the depth of each rack are known in advance.

$$\min \sum_{j=1}^{n_r} (C_v + C_{rw} whlnk_j + C_j^{st}) \tag{J.34}$$

$$\text{s.t.} \sum_{j=1}^{n_r} y_{ij} = Q_i, \qquad i \in I, \tag{J.35}$$

$$\sum\nolimits_{i \in I} n_{ij} \leq n, \qquad 1 \leq j \leq n_r, \tag{J.36}$$

$$d_j \geq C_s + C_t \sum_{i \in I} \frac{d_i y_{ij}}{Q_i} \left(\frac{1}{n} EC_{k_j}^1 + \left(1 - \frac{1}{n}\right) EC_{k_j}^2 \right), \qquad 2 \leq k, 1 \leq j \leq n_r, \tag{J.37}$$

$$e_j \geq C_s n + C_t \sum_{i \in I} \frac{d_i y_{ij}}{Q_i} EC_{k_j}^1, \qquad 2 \leq k, 1 \leq j \leq n_r, \tag{J.38}$$

$$d_j - e_j \leq M_2 f_j, \qquad 1 \leq j \leq n_r, \tag{J.39}$$

$$e_j - d_j \leq M_2 (1 - f_j), \qquad 1 \leq j \leq n_r, \tag{J.40}$$

$$C_j^{st} \geq e_j - M_2 (1 - f_j), \qquad 1 \leq j \leq n_r, \tag{J.41}$$

$$C_j^{st} \geq d_j - M_2 f_j, \qquad 1 \leq j \leq n_r, \tag{J.42}$$

$$k_j n_{ij} - y_{ij} \geq 0, \qquad i \in I, 1 \leq j \leq n_r, \tag{J.43}$$

$$y_{ij}, n_{ij} \in Z^+, d_j, e_j, f_j \in R^+.$$

Appendix K. Comparison between The Standard and The Modified B&B Algorithm

Except for the 10 cases shown in Table 4-2. Comparison of Solving Methods, we created another 50 cases to makes sure that the modified B&B algorithm can at least, with a high probability to reach the optimal solution, and generate a much smaller branch-and-bound tree. Surprisingly, as shown in Table S-4. Comparison of the Standard and the Modified B&B Algorithm, the modified B&B algorithm can always reach the optimal solutions, which indicates the accuracy of the two properties.

Table S-4. Comparison of the Standard and the Modified B&B Algorithm

$[n, \lvert I \rvert, \sum_{i \in N} Q_i, C_t]$	$C_{B\&B}$	$C_{B\&B'}$	$T_{B\&B}$	$T_{B\&B'}$	$NON_{B\&B}$	$NON_{B\&B'}$
[16, 14, 161, 61]	927	927	6	5	177	135
[16, 16, 192, 66]	1005	1005	23	14	204	192
[22, 20, 246, 21]	634	634	14	3	108	52
[12, 9, 84, 83]	895	895	2	0	70	50
[16, 16, 210, 59]	1007	1007	90	45	350	323
[18, 16, 195, 119]	1394	1394	62	55	449	245
[23, 18, 240, 9]	497	497	3	2	112	90
[17, 19, 214, 116]	1438	1438	1300	708	677	269
[11, 13, 176, 122]	1474	1474	301	166	1197	749
[16, 14, 144, 67]	926	926	3	2	102	88
[9, 12, 166, 28]	771	771	54	30	565	513
[7, 5, 39, 54]	588	588	0	0	25	15
[25, 20, 249, 93]	1269	1269	8	7	143	118
[13, 14, 173, 107]	1315	1315	74	44	621	335
[8, 8, 127, 47]	850	850	8	4	563	366
[8, 8, 101, 12]	451	451	4	2	107	56
[14, 13, 170, 39]	778	771	5	3	185	130
[18, 18, 231, 43]	876	876	15	14	232	186
[14, 12, 139, 36]	682	682	3	2	230	84
[4, 7, 90, 75]	1167	1167	129	117	4213	3130
[23, 24, 480, 83]	1565	1565	2100	1739	16502	1786
[26, 31, 372, 59]	1175	1175	35	21	482	376
[36, 31, 372, 119]	1660	1660	93	88	192	149
[38, 33, 627, 9]	752	752	83	59	416	289
[36, 34, 612, 116]	1967	1967	1288	1071	1429	1157
[26, 28, 504, 122]	1935	1935	2849	1746	4296	2132
[30, 29, 580, 67]	1432	1432	530	528	1283	1281
[28, 27, 378, 28]	836	836	25	14	276	225
[17, 20, 200, 54]	880	880	10	7	170	142
[20, 24, 579, 83]	1814	1814	117380	11617	31058	12204
[27, 31, 482, 59]	1308	1308	1174	582	1072	915
[31, 31, 396, 119]	1695	1695	393	329	547	264
[33, 33, 712, 9]	830	830	991	625	1437	699
[27, 29, 477, 61]	1325	1325	1751	750	2384	930
[35, 31, 545, 66]	1386	1386	863	260	1340	694
[35, 35, 711, 21]	1035	1035	654	28	2432	624

Table S-5. Comparison of the Standard and the Modified B&B Algorithm (Cont'd)

[28, 24, 326, 83]	1303	1303	176	83	242	240
[33, 31, 505, 59]	1295	1295	1085	263	3042	561
[28, 31, 496, 119]	1857	1857	11053	1153	4075	1062
[31, 33, 557, 9]	725	725	355	151	961	129
[34, 34, 640, 116]	1998	1998	13031	3017	7503	1718
[25, 28, 492, 122]	1948	1948	5465	2537	4339	1982
[30, 29, 506, 67]	1418	1418	792	577	2495	762
[28, 27, 503, 28]	991	991	237	199	857	643
[24, 20, 323, 54]	1053	1053	66	14	373	284
[35, 35, 597, 93]	1760	1760	1282	604	956	528
[34, 29, 534, 107]	1795	1795	1599	739	1247	804
[27, 23, 414, 47]	1104	1104	636	167	622	236
[22, 23, 365, 12]	652	652	113	11	825	134
[27, 28, 456, 39]	1072	1072	254	237	1379	528
[34, 33, 626, 43]	1231	1231	643	585	2776	844
[26, 27, 470, 36]	1063	1063	140	127	752	700
[21, 22, 339, 75]	1342	1342	284	278	3077	890

Appendix L. Sensitivity Analysis of Inventory Levels and Demand Rates Distribution

Table S-6. Sensitivity Analysis on Inventory Levels and Demand Rates Distribution when $C_t = 1$

$G(0.2)$	σ	$C_{B\&B}$	EC	$[k_1, k_2, .., k_{n_r}]$	$[m_1, m_2, .., m_{n_r}]$	S	Avg	std
0.2	1	512	42.8	[26]	[1]	754	26.0	0.0
	6	584	49.5	[31]	[1]	899	31.0	0.0
	11	612	24.7	[10, 15]	[1, 1]	725	12.5	2.5
	16	625	25.0	[9, 17]	[1, 1]	754	13.0	4.0
	21	625	25.3	[6, 20]	[1, 1]	754	13.0	7.0
0.4	1	512.	42.8	[26]	[1]	754	26.0	0.0
	6	584	49.5	[31]	[1]	899	31.0	0.0
	11	612	25.1	[9, 16]	[1, 1]	725	12.5	3.5
	16	626	25.7	[10, 16]	[1, 1]	754	13.0	3.0
	21	628	28.2	[6, 20]	[1, 1]	754	13.0	7.0
0.6	1	512	42.8	[26]	[1]	754	26.0	0.0
	6	584	49.5	[31]	[1]	899	31.0	0.0
	11	611	23.9	[9, 16]	[1, 1]	725	12.5	3.5
	16	625	24.9	[9, 17]	[1, 1]	754	13.0	4.0
	21	628	27.9	[6, 20]	[1, 1]	754	13.0	7.0
0.8	1	512	42.8	[26]	[1]	754	26.0	0.0
	6	584	49.5	[31]	[1]	899	31.0	0.0
	11	611	23.9	[9, 16]	[1, 1]	725	12.5	3.5
	16	625	24.9	[9, 17]	[1, 1]	754	13.0	4.0
	21	628	27.9	[6, 20]	[1, 1]	754	13.0	7.0

Table S-7. Sensitivity Analysis on Inventory Levels and Demand Rate Distribution when $C_t = 25$

$G(0.2)$	σ	$C_{B\&B}$	EC	$[k_1, k_2, .., k_{n_r}]$	$[m_1, m_2, .., m_{n_r}]$	S	Avg	std
0.2	1	1108	14.1	[7, 8, 9]	[29, 29, 29]	696	8.0	0.81
	6	1105	14.0	[7, 8, 9]	[29, 29, 29]	696	8.0	0.81
	11	1103	13.9	[7, 8, 9]	[29, 29, 29]	696	8.0	0.81
	16	1093	13.5	[4, 9, 11]	[29, 29, 29]	696	8.0	2.94
	21	1080	12.9	[4, 9, 11]	[29, 29, 29]	696	8.0	2.94
0.4	1	1100	13.8	[7, 8, 9]	[29, 29, 29]	696	8.0	0.8
	6	1102	13.9	[7, 8, 9]	[29, 29, 29]	696	8.0	0.81
	11	1104	13.9	[7, 8, 9]	[29, 29, 29]	696	8.0	0.81
	16	1103	13.9	[6, 8, 10]	[29, 29, 29]	696	8.0	1.6
	21	1104	13.9	[5, 9, 10]	[29, 29, 29]	696	8.0	2.2
0.6	1	1071	12.6	[5, 7, 12]	[29, 29, 29]	696	8.0	2.94
	6	1081	13.0	[5, 7, 12]	[29, 29, 29]	696	8.0	2.94
	11	1082	13.1	[5, 8, 11]	[29, 29, 29]	696	8.0	2.44
	16	1088	13.3	[6, 8, 10]	[29, 29, 29]	696	8.0	1.63
	21	1088	13	[5, 9, 10]	[29, 29, 29]	696	8.0	2.16
0.8	1	1010	10.8	[3, 8, 13]	[29, 29, 29]	696	8.0	4.08
	6	1021	11.3	[3, 9, 12]	[29, 29, 29]	696	8.0	3.74
	11	1024	10.7	[2, 9, 13]	[29, 29, 29]	696	8.0	4.54
	16	1034	11.1	[3, 9, 12]	[29, 29, 29]	696	8.0	3.74
	21	1055	12.6	[5, 9, 10]	[29, 29, 29]	696	8.0	2.16

Table S-8. Sensitivity Analysis on Inventory Levels and Demand Rate Distribution when $C_t = 50$

$G(0.2)$	σ	$C_{B\&B}$	EC	$[k_1, k_2, .., k_{n_r}]$	$[m_1, m_2, .., m_{n_r}]$	S	Avg	std
0.2	1	1460	14.1	[7, 8, 9]	[29, 29, 29]	696	8.0	0.81
	6	1455	14.0	[7, 8, 9]	[29, 29, 29]	696	8.0	0.81
	11	1451	13.9	[7, 8, 9]	[29, 29, 29]	696	8.0	0.81
	16	1431	13.5	[4, 9, 11]	[29, 29, 29]	696	8.0	2.94
	21	1404	12.9	[4, 9, 11]	[29, 29, 29]	696	8.0	2.94
0.4	1	1445	13.8	[7, 8, 9]	[29, 29, 29]	696	8.0	0.8
	6	1449	13.9	[7, 8, 9]	[29, 29, 29]	696	8.0	0.81
	11	1454	13.9	[7, 8, 9]	[29, 29, 29]	696	8.0	0.81
	16	1451	13.9	[6, 8, 10]	[29, 29, 29]	696	8.0	1.6
	21	1453	13.9	[5, 9, 10]	[29, 29, 29]	696	8.0	2.2
0.6	1	1271	12.6	[3, 8, 13]	[29, 29, 29]	696	8.0	2.94
	6	1287	13.0	[3, 9, 12]	[29, 29, 29]	696	8.0	2.94
	11	1294	13.1	[2, 9, 13]	[29, 29, 29]	696	8.0	2.44
	16	1314	13.3	[3, 9, 12]	[29, 29, 29]	696	8.0	1.63
	21	1356	13	[5, 9, 10]	[29, 29, 29]	696	8.0	2.16
0.8	1	1271	10.8	[3, 8, 13]	[29, 29, 29]	696	8.0	4.08
	6	1287	11.3	[3, 9, 12]	[29, 29, 29]	696	8.0	3.74
	11	1294	10.7	[2, 9, 13]	[29, 29, 29]	696	8.0	4.54
	16	1314	11.1	[3, 9, 12]	[29, 29, 29]	696	8.0	3.74
	21	1055	12.6	[5, 9, 10]	[29, 29, 29]	696	8.0	2.16

Table S-9. Sensitivity Analysis on Inventory Levels and Demand Rate Distribution when $C_t = 100$

$G(0.2)$	σ	$C_{B\&B}$	EC	$[k_1, k_2, .., k_{n_r}]$	$[m_1, m_2, .., m_{n_r}]$	S	Avg	std
0.2	1	2201	10.0	[4, 4, 5, 5, 6]	[29, 29, 29, 29, 29]	696	4.8	0.7
	6	2188	9.8	[4, 4, 5, 5, 6]	[29, 29, 29, 29, 29]	696	4.8	0.7
	11	2163	9.6	[3, 4, 5, 6, 6]	[29, 29, 29, 29, 29]	696	4.8	1.1
	16	2139	9.4	[3, 4, 5, 5, 7]	[29, 29, 29, 29, 29]	696	4.8	1.3
	21	2101	9.1	[2, 4, 5, 6, 7]	[29, 29, 29, 29, 29]	696	4.8	1.7

0.4	1	2160	9.6	[3, 4, 5, 6, 6]	[29, 29, 29, 29, 29]	696	4.8	1.1
	6	2180	9.8	[4, 4, 5, 5, 6]	[29, 29, 29, 29, 29]	696	4.8	0.7
	11	2177	9.8	[3, 5, 5, 5, 6]	[29, 29, 29, 29, 29]	696	4.8	0.9
	16	2177	9.8	[3, 4, 5, 6, 6]	[29, 29, 29, 29, 29]	696	4.8	1.1
	21	2170	9.7	[3, 4, 5, 6, 6]	[29, 29, 29, 29, 29]	696	4.8	1.1
0.6	1	2067	10.1	[3, 6, 7, 8]	[29, 29, 29, 29, 29]	696	6.0	1.8
	6	2099	10.4	[3, 6, 7, 8]	[29, 29, 29, 29, 29]	696	6.0	1.8
	11	2110	9.2	[2, 4, 5, 6, 7]	[29, 29, 29, 29, 29]	696	4.8	1.7
	16	2124	9.3	[3, 4, 5, 6, 6]	[29, 29, 29, 29, 29]	696	4.8	1.1
	21	2136	9.4	[3, 4, 5, 6, 6]	[29, 29, 29, 29, 29]	696	4.8	1.1
0.8	1	1878	8.6	[1, 4, 7, 12]	[0, 29, 29, 29]	696	6.0	4.0
	6	1916	8.8	[2, 5, 8, 9]	[29, 29, 29, 29, 29]	696	6.0	2.7
	11	1930	8.9	[2, 6, 7, 9]	[29, 29, 29, 29, 29]	696	6.0	2.5
	16	1980	9.3	[2, 4, 8, 10]	[29, 29, 29, 29, 29]	696	6.0	3.1
	21	2007	9.6	[3, 6, 7, 8]	[29, 29, 29, 29, 29]	696	6.0	1.8

VITA

Wenquan Dong was born in Shandong, China. He graduated in 2016 with a Bachelor's degree in Construction Management from Hohai University, China. In the fall of 2017, he joined the Department of Industrial and Systems Engineering of the University of Tennessee, Knoxville (UTK), and started to work on the design and operation of automated warehouse systems with Dr. Mingzhou Jin. In the fall of 2018, he started to work with Dr. Sachin U Nimbalkar, Dr. Wei Guo, and Kristina Armstrong on the estimation of the sustainability of the U.S. Food System. He is expected to complete his Doctor of Philosophy degree in 2021. His research interests include warehouse design and operation, design of the last-mile delivery network with drones, and sustainability of food supply chains.