Facility layout and emerging advanced material handling optimization

by

Hung-Yu Lee

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
August 5, 2017

Keywords: Facility layout problem, vehicle routing problem, warehouse operation, mathematical optimization

Approved by

Dr. Chase Murray, Chair, Affiliate Professor of Industrial and Systems Engineering
Dr. Andres Carrano, Philpott-WestPoint Stevens Associate Professor of Industrial and Systems Engineering
Dr. Alice Smith, Joe W. Forehand/Accenture Distinguished Professor of Industrial and Systems Engineering
Dr. Jeffrey Smith, Joe W. Forehand Jr. Professor of Industrial and Systems Engineering

Abstract

This research explores the intersection of facility layout and vehicle routing, beginning with a study on a real-world semiconductor manufacturing facility layout problem (FLP), followed by a study of facility layout alternative analysis for the implementation of autonomous warehousing robots, and concluding with a research of the vehicle routing problems (VRPs) for the warehousing robots. The first study proposes a new type of FLP with space utilization concerns to determine if the existing facility is large enough to accommodate the machines required by a suite of capacity contingency plans. New efficient heuristic solution procedures have been developed to help our industry partner identify the facility layout alternatives for future demand satisfaction. The second study seeks to determine if any common warehouse layouts that work well with only human pickers are also preferable when novel autonomous warehousing systems are deployed. In order to answer the question, the warehousing behavior of the autonomous system is modeled as a vehicle routing problem (VRP) with multiple synchronization constraints. Numerical experiments are conducted and analyzed to provide guidelines for warehouse design under the proposed routing systems. The Third study proposes an efficient heuristic to solve two related routing problems for the autonomous warehousing systems. Numerical analyses conducted indicate that the heuristic is able to provide optimal or near-optimal solutions in a reasonable computing time for practical situations, and demonstrate the benefit associated with the proposed routing model. In conclusion, managerial insights of these three studies and the potential future research are discussed.

Acknowledgments

This dissertation stands as a testament to the support and encouragement from my family, friends, and colleagues. First of all, I would like to acknowledge the academic environment with generous supports on facilities and funding provided by Auburn University and the Department of Industrial & Systems Engineering at Auburn University.

My sincere thanks goes to my committee members, Dr. Andres Carrano, Dr. Alice Smith, and Dr. Jeffrey Smith. Their brilliant comments and suggestions have been obvious throughout this dissertation. I would also like to thank Dr. Saad Biaz for the willingness to serve as my university reader.

I would like to express my deepest appreciation to my advisor, Dr. Chase C. Murray, who is not only a tremendous mentor but also a friend in my life. Dr. Murray has supported my studies with his comments, remarks and involvement, and has trained me to be a better scientist/researcher/engineer. This dissertation would have never been accomplished without his knowledge and guidance. I cannot say thank him enough for his patience and encouragement. It has been a great honor to work for a person of his brilliance, and I will definitely miss our weekly meetings.

I also wish to show my sincere gratitude to faculty members who have contributed my professional experience. In particular, I would like to thank Dr. James Chen, Dr. Jiun-yan Shiau, Dr. Kung-Jeng Wang, Dr. Ling-Huey Su, and Dr. Kuo-Hwa Chang. They have prepared, encouraged and supported me to pursue advanced education and further. With a special thank to Dr. Mark Schall, and Dr. Jeffrey Smith, for their commitment to teaching. It was a pleasure to be in their classrooms and to serve as their teaching assistant.

I am deeply appreciative of the cares from friends who have been there to support me getting through the hard times. I thank Mike Wang, Daphne Ku, Yichuan (Eason) Wang, Erik Huang, Yen-Chen (Leeann) Juan, Barry Yeah, and Derrong Irene Lin, for looking out my best interests since I arrived Auburn, for providing an abundance of suggestions as well as countless dinner parties and workout sections. I also thank Eren Sakinc, Joseph Ekong, Rahul Reddy, Ali Khoshkhoo, Volkan Barim, Menekse Salar, Nasrin Mohabbati, and Zahra Sedighi, for their personal and professional support during the time I spent at the University. I cannot express my gratitude enough for their friendship. I am also grateful to Dr. Chaochi Ting for his kindness to support me during my service as the president of the INFORMS student chapter.

Finally, I wish to thank my family: my parents, Mau-Rung Lee and Hai-Tsai Lin Lee, and my uncle, Man-Shiang Lee for supporting me spiritually and financially while my time during Ph.D. study and my life in general. They give me the encouragement to pursue my goals. I would also like to thank my uncle and aunt, Chris Li and Kelly Li, for their kindness and support. This dissertation would not be completed without the supports of my family. I thank them all for the many prayers they have said on my behalf, and I will be grateful for their love forever.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This research addresses the intersection of facility layout and vehicle routing problems. Three distinct research topics are addressed in this dissertation, beginning with a study on a real-world facility layout problem (FLP), followed by an investigation of the impacts of facility layout on autonomous item-picking robots, and concluding with a study of the vehicle routing problems (VRPs) as the extension of the second topic.

The first research topic, "Maximizing Space Utilization in Semiconductor Manufacturing Facilities," focuses on a FLP for semiconductor manufacturing systems from the perspective of space utilization. Traditional FLPs seek to minimize the cost of material handling, which is measured as the total distance travelled by each product through the manufacturing process. In such problems, the determination of where to place each machine is a decision based on current product demands. In this research, a new type of FLP with space utilization concerns for semiconductor manufacturing wafer fabrication facilities (fabs) is proposed to analyze the feasibility of fitting the required machine sets into an existing facility. This research is motivated by a real-world challenge faced by a semiconductor manufacturer with long-term capacity planning uncertainty. In particular, this manufacturer seeks to determine if the existing facility is large enough to accommodate the machines required by a suite of capacity contingency plans. The resulting problem of maximizing space utilization is complicated by the required "functional area" layout structure employed by our industrial partner. This layout structure allows clearance space sharing for adjacent machines and non-rectangular functional areas. We have developed mathematical programming formulations, which are an amalgamation of multiple NP-hard problems for the resulting facility layout problems.

New efficient heuristic solution procedures have been developed to help our industry partner identify the facility layout alternatives for future demand satisfaction.

In the second topic, "Robotics in order picking: Evaluating warehouse layout for pick, place, and transport-vehicle routing systems," a novel routing problem of a new autonomous warehouse order picking system is presented. The problem was inspired by the development of warehousing robotics: Fetch and Freight. An item picking robot hand, a Fetch, picks up an item from the shelf then places it into the tote on a Freight, which is an autonomous vehicle transiting items from each Fetch to the depot. The routing problem is modeled as a variant of VRP with multiple synchronization constraints, which seeks to minimize the makespan associated with delivering all items from a batch pick list to the packing station. The problem termed the pick, place, and transport- vehicle routing problem (PPT-VRP). A mixed integer linear programming formulation is developed to answer two related research questions. First, what combination of picker and transport robots is required to obtain performance exceeding traditional human-based picking operations, where human workers pick items and return the entire batch to the packing station manually? Second, if it is possible to expand the fleet of robots or to enhance the capabilities of the current fleet, which type of additional robot or enhancement leads to the greatest performance improvement? Furthermore, this research seeks to determine if warehouse layouts that work well with only human pickers are also preferable when coordinated mobile robots are deployed. In conclusion, numerical experiments are conducted and analyzed to provide guidelines for warehouse design under the proposed routing systems.

An extension of the PPT-VRP, "Heuristic Approaches for Advanced Pick, Place, and Transport vehicle Routing Optimization Problems: Applications in Warehouse Order Picking Robotics," is studied as the third topic. Although the PPT-VRP assumes a picker can only pass the held item to a transporter at the picking location, the performance of order picking could be improved if pickers can pass items to transporters at any locations of the warehouse. Thus, an advanced routing model of the PPT-VRP (APPT-VRP)is proposed to address this situation. A heuristic approach for solving the model is developed because of the excessive computational time required in our preliminary results by existing MILP solvers. Numerical analyses conducted indicate that the newly proposed heuristic approach is able to provide near-optimal solutions in a reasonable computing time for practical situations. In order to provide managerial insight of this study, we investigate the overall performance for the two formulations. According to the results, the APPT-VRP is not only superior to the PPT-VRP in terms of average makespan, but also yields a smaller variance, which provides better service level in terms of overall response time for warehousing under uniform storage policy.

Chapter 2

Maximizing Space Utilization in Semiconductor Manufacturing Facilities

2.1 Introduction

This research is motivated by a problem experienced by a semiconductor foundry located in the northeastern United States with whom we have collaborated. Whereas semiconductor manufacturers such as Intel design and produce their own products in-house, foundries fabricate the designs of other companies (which then sell these products to third parties). One of the challenges faced by foundries in particular is that long-range production planning forecasts are difficult to determine, as the foundry does not maintain its own product line. Owing to the high cost of cleanroom space required in semiconductor manufacturing facilities (known as fabs), as well as the long lead times associated with altering production capacity, semiconductor manufacturers are required to plan far in advance to ensure tool quantities that are sufficient to accommodate dynamic demands. As fab construction costs can exceed $4,000 per square foot, with annual operating costs of $750 per square foot [26], adding square footage to a facility is an expensive proposition.

Our industrial partner approached us with a simple question: *Do we have enough space in our existing facility to house all of the machines required for a particular capacity plan?* Although this is a straightforward question, a number of constraints lead to a new and challenging class of facility layout problem. First, machines in fabs are generally grouped according to functional areas (or modules). Each functional area (FA) contains machines that perform similar processes, such as photolithography, etching, or implantation. This configuration is common due to the complex re-entrant flow required to fabricate a semiconductor, such that each wafer may revisit each functional area multiple times. Because a fab may produce a number of different products, each requiring a unique process flow, functional area layouts are the standard layout configuration across the industry.

For the capacity planning in the industry, Chen et al. [6] proposed a fab design procedure to develop and evaluate fab design alternatives based on availability of both machines and automated material handling systems (AMHS). The procedure considers the pre-determined material flows and distances among FAs, the efficiency of the AMHS, and the capacities of machines to estimate required machines in each FA based on the future demands in order to provide and to evaluate facility design alternatives in the initial phase of fab design. Due to the complexity of the facility layout problem for a whole facility in practice, the relative location of FAs is pre-determined in order to provide the estimates of material flows among FAs for the capacity planning. Thus, the common material handling flow objective for facility layout research is incorporated in pre-determined relative location of FAs, so in our research, the objective is to minimize total space consumption when considering the total number of machines required for a particular capacity contingency plan, column sharing for FAs, column space sharing for machines, and clearance requirements between machines. Our particular problem needs to consider the overall (outer) dimensions of the facility, the height of each row, and the width of the walkways separating the rows, all of which are pre-determined (given). Furthermore, in the case of the particular question posed by our industry partner, the sequence of functional areas assigned to each row is given. However, the size of each functional area in each row (determined by the allocation of machines to each bay and the number of bays/walkways required) is unknown. Moreover, the proposed methodology is applicable for the situation when the relative location of FAs is flexible, because in practice, some semiconductor manufacturers may have their own production process and their own way to manage contamination to make the relative location flexible. In such case, given an initial relative location of FAs, the proposed method searches for better layout alternatives by relaxing the initial relative locations. Details will be introduced later.

To introduce the facility layout in details, within the fab, functional areas may be distributed across multiple rows of differing heights, as shown in Figure 1. Each row contains contiguous groups of FAs, and may be of differing "heights." Note that some functional areas may be assigned to multiple rows (e.g., the "films" FA appears in all three rows in Figure 2.1). Thus, this problem requires us to not only locate the machines in each FA, but also to determine to which replicate of that FA each machine should be allocated.

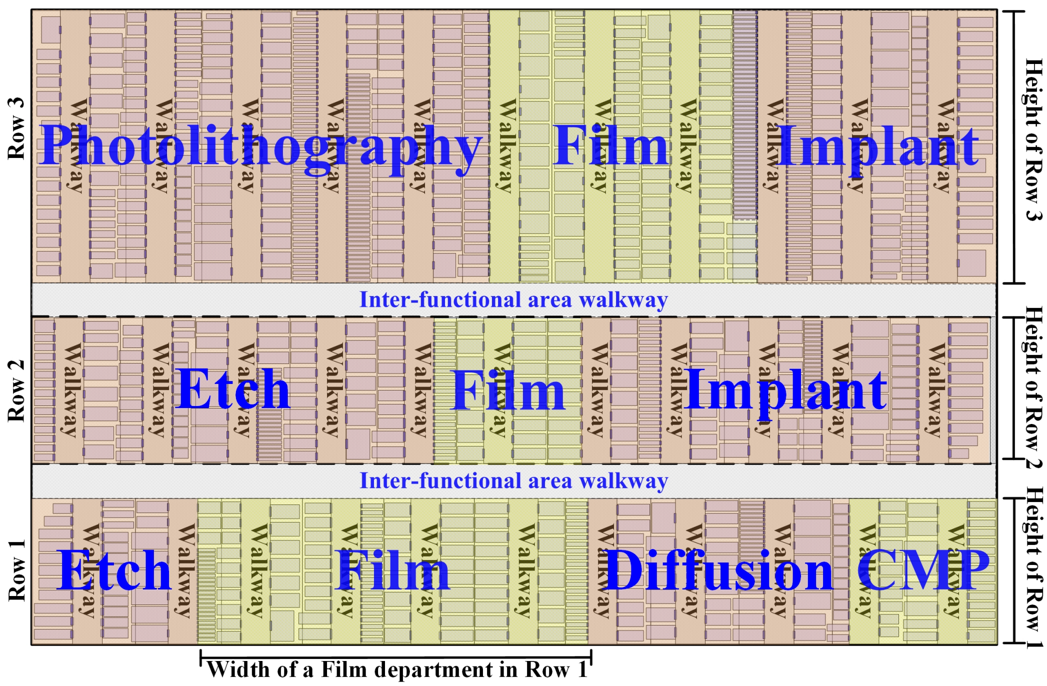


Figure 2.1: A fab layout showing FAs located in three rows.

Space utilization is improved by allowing adjacent FAs within the same row to share a machine column on the boundaries, which we term "FA interlocking." For example, the films and implant FAs located in the top row of Figure 2.1 are non-rectangular interlocking FAs. Within each FA, a "bay" structure is common, whereby a walkway separates two parallel "columns" of machines, as shown in Figure 2.2. Cassettes of wafers are loaded into the ports of machines, which are accessible from the walkway. Similar to the concept of FA interlocking, space can also be conserved by allowing "machine interlocking," whereby machines in back-to-back columns may extend into the adjacent bay. This interlocking must observe the pre-determined clearance requirements between machines, including side and back clearances, to allow sufficient ventilation and access to the machines for maintenance. We term this the multi-row multi-interlocking-column (MRMIC) facility layout configuration.

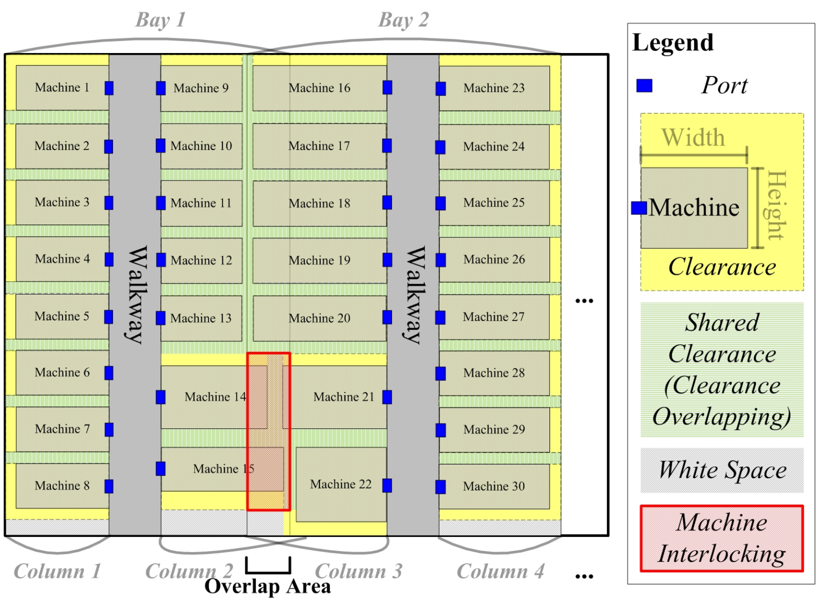


Figure 2.2: Detailed view within a FA, showing a layout with interlocking machines.

Although the MRMIC layout structure is widely applied in the semiconductor manufacturing industry, we are not aware of any published studies in which this particular structure was examined. Furthermore, no facility layout design problems have considered space utilization as a primary objective function. This paper provides a mathematical programming formulation with corresponding heuristics in an attempt to solve the resulting facility layout problem, which is an amalgamation of multiple NP-hard problems, including different types of facility layout problems (FLPs) and bin packing problems. Unlike existing FLP literature, large-scale problems with more than 1000 machines are considered as a test bed to measure the performance of the proposed approach. Because we were unable to find any applicable approaches to solving our problem in the existing literature on FLPs and bin packing, a lower bound is proposed to assess the quality of the resulting solutions.

This paper is organized as follows: A review of related literature is provided in Section 2.2. A formal problem description, including mathematical programming formulations, is given in Section 2.3. The proposed solution approach is described in Section 2.4, followed by a numerical analysis in Section 2.5. Finally, concluding remarks and opportunities for further research are presented in Section 2.6.

2.2 Related literature

Among the myriad streams of facility layout literature, the unequal size (block) FLP – which seeks to minimize material flow costs by allocating unequal-sized departments into a given floor space – is most closely related to the problem of interest. A variety of mathematical programming formulations have been provided for this problem, including Castillo and Westerlund [3], Castillo et al. [4], and Taghavi and Murat [25]. The bay block FLP, which has been applied in determining the relative location of FAs in semiconductor manufacturing, has been studied by Chae and Peters [5], Chen et al. [7], Konak et al. [16], and Kulturel-Konak [17]. This problem focuses on locating departments in parallel bays of varying heights that are bounded by straight aisles on both sides and are connected at the ends by AMHS. The variants of the FLP with stochastic material flows are considered in Kulturel-Konak et al. [18], Norman and Smith [20], and the problem with deterministic flows but with heterogeneous from/to distance metrics is discussed inOzdemir et al. [22]. In addition, the problems with determining numbers and locations of input/output (I/O) points were discussed by Arapoglu et al. [2] and Norman et al. [21]. Most block FLPs seek to minimize material flow costs in layouts with departments in parallel bays, but the shapes of departments are flexible or pre-determined with the same widths. The proposed MRMIC layout design extends the block FLP by considering space sharing for differently-shaped machines between two machine columns in each bay, walkways between every two columns, clearance requirements between machines, and an objective of minimizing total space requirements.

A number of fab layout design studies have addressed the use of AMHS with uni-directional (bi-directional) tracks to minimize the cost of material flow between FAs. For example, Peters and Yang [23] solved an integrated facility and material handling system layout design problem which considered spine and perimeter configurations with two types of material handling patterns. An integer programming (IP) model with a branch and bound approach to solve a spine semiconductor layout design problem with a given unidirectional AMHS was proposed by Yang and Peters [29], while Yang et al. [30, 31] have proposed department fab layout design problems with a fixed two-way spine AMHS. Yang et al. [32] proposed a fuzzy logic methodology using linguistic variables to evaluate 10 fab layout alternatives. Several approaches, such as systematic layout planning (SLP), analytic hierarchy process (AHP) for multiple objectives, integer linear programming (ILP), simulated annealing (SA), and tabu search (TS) have been applied to solve fab layout design problems. However, these only dealt with the problem on a functional area level and assumed all the FAs are a rectangular shape. Although Ueda et al. [27] proposed an open space layout design problem at the machine level with non-track automatic guided vehicles (AGV), this required much more floor space than using a spine layout design with AMHS. In addition, due to the complexity of the problem, only small-scale problems with less than 150 departments have been tested using these methods.

Also related to our study is the double row layout problem (DRLP), which seeks the arrangement of machines on two sides of a single corridor to minimize material handling costs, subject to minimum clearance requirements between machines. A mathematical programming formulation was proposed and corrected by Chung and Tanchoco [9] and Zhang and Murray [34], respectively. Due to the cost of floor space, multi-objective DRLPs have been proposed with minimizing space requirements as one of the objectives by Murray et al. [19], Zuo et al. [35]. To further improve space utilization and material handling cost, the concept of shared clearances between adjacent machines was proposed in Zuo et al. [36]. Unlike the DRLP, the present work considers multiple double-row layouts simultaneously.

Simulation models have been developed to assess the performance of alternative layout designs. For example, Geiger et al. [12] compared five layout alternatives in simulation with differing quantities of machines. Several levels of availability, setup time, and transfer time were evaluated according to the resulting cycle time. Chung and Jang [8] compared the integrated room layout (IRL) design with the traditional spine layout design, and the results showed that IRL dominated spine design in several respects, such as process flexibility, cycle time, and space usage. While these simulation models provide a way to compare alternatives that were generated by experience and intuition, they are unable to identify the optimality of a particular layout alternative.

Beyond the FLP, the present paper also features aspects of bin packing problems, which aim to optimize space utilization by packing a collection of items within a number of given-sized bins. Bin packing problems with different considerations were reviewed in Delorme et al. [10]. Recent proposed upper and lower bounds for bin packing problems were surveyed in Zhang et al. [33]. A number of solution approaches for bin packing problems have been proposed, including greedy random local search [1], memetic algorithm [11], maximum rectangular rest space arrangement [13], edge score with search algorithms [14, 33], and a goal-driven approach with skyline representation [28]. Because these bin packing problems assume that the bin sizes are given, solution approaches for these problems are not directly applicable to the proposed problem, in which sizes and shapes of the FAs and tool columns are flexible.

A summary of the related literature is provided in Table 2.1, which highlights the key features of our research and the scale of test problems evaluated. This paper makes several contributions to the literature. First, motivated by real problems to determine the detailed facility layout alternatives of fitting the required machine set , which is based on the results of the capacity planning process, into expensive and existing facility. This research is the first to focus on maximizing space utilization in terms of minimizing required floor space in FLP to address the problems. Second, unlike the DRLP, which only considers layout design within

Table 2.1: A comparison of related literature

Research	Reference	Objective Function	Problem Scale	Configuration
Bin Packing Problem	[1] [11] [13] [14] [28] [33]	Number of Used Bins, Total Used Bin Costs, ...	$\leq$ 1,000 $\leq$ 2,900 ([28])	Objects in Given Size Bins
Block FLP	[3] [4] [25]	Material Flow Costs	$\leq$ 20	Open Space
DRLP	[9] [19] [23] [30] [31] [34] [35] [36]	Material Flow Costs w/o Space Requirement	$\leq$ 50	Double Row
FLP for Semiconductor Manufacturing	[23] [27] [29] [30] [31]	Material Flow Costs	$\leq$ 150	Double Row / Open Space
Layout Analysis	[8][12]	Material Flow Costs, Flexibility, Cycle Time ...	$\leq$ 500	Multiple-Double Row
Parallel-Bay Block FLP	[2] [5] [7] [16] [17] [18] [20] [21] [22]	Material Flow Costs	$\leq$ 60	Open Space with Single Column Department in each Bay
This Research	-	Space Requirement	$\geq$ 1,500	MRMIC Layout

one double-row, this research implements the MRMIC layout structure for facilities with multiple double-rows. Although DRLP has practical values by determining the locations of FAs within two rows in the industry, it cannot be applied to the proposed problem. This paper presents a new mathematical programming formulation for this new space-minimizing facility layout problem with non-rectangular FAs and space sharing for adjacent columns. Finally, an efficient approach for solving large-scale instances of this problem is proposed.

2.3 Problem definition

The MRMIC facility layout design problem considers the allocation of machines into parallel columns within parallel functional area rows. We define and formulate the restricted version of the problem, which is the relative location of the FAs in each row is given (as was the case with our industrial partner and for the most of the manufacturers), although the width of each FA is flexible.

Consistent with the DRLP, the loadport side of machines should be lined up on the left or right edge of walkways. Each walkway between adjacent columns has a fixed width, W^P, but the total number required walkways is unknown *a priori*. Unlike [36], which only considered clearance sharing on the sides of machines, the back clearances can also be shared in the proposed problem. The back clearance of a machine is on the opposite side of the loadport, while the side clearance is on the two flank sides. The minimum clearance of each type between two machines must be at least the maximum clearance requirement of either machine.

Several parameters and variables are defined to formulate this problem. For parameters, let W^F represents the width of the floor. A layout plan is only feasible if the required width to fit all targeted machines does not exceed W^F. Let $\mathcal{R}$ represent the set of rows, where $\mathcal{R} = \{1, 2, \ldots\}$. We define H_r^R to represent the available height of row r, where $r \in \mathcal{R}$. In order to make sure machines are located within a row, we define Y_r^L (Y_r^U) to represent the lower (upper) edge y-coordinates of row r, where $r \in \mathcal{R}$. Next, let $\mathcal{F}$ represent the set of functional areas, and we define $\Phi_{f,r}$ to represent the order of functional area f in row r from left to right. For example, if functional area f is the third functional area from the left in row r, then $\Phi_{f,r} = 3$. Note if functional area f does not appear in row r then $\Phi_{f,r} = 0$, where $f \in \mathcal{F}$ and $r \in \mathcal{R}$. Moreover, let $\mathcal{M}_f$ represent the set of machines from functional area f, where $f \in \mathcal{F}$, and let H_m^M (W_m^M) represent the height (width) of machine i, where $m \in \mathcal{M}_f$, and $f \in \mathcal{F}$. For determining the clearance requirement, let $C_{m,n}^{M,B}$ ($C_{m,n}^{M,S}$) represent the required back (side) clearance between machine m and machine n. $C_{m,o}^{M,b}$ ($C_{m,o}^{M,S}$) represents the back (side) clearance requirement of machine m (also the clearance requirement between machine m and the wall). We can calculate $C^{M,B}_{m,n}$ =max$\{C^{M,B}_{m,o}, C^{M,B}_{n,o}\}$, where $m, n \in \mathcal{M}_f$, $f \in \mathcal{F}$, and $m \neq n$. The same rule applies to $C^{M,S}_{m,n}$. Additionally, let $\mathcal{P}$ represent the possible walkways in each row, where $\mathcal{P} = \{1, 2, \ldots, P^{\max}\}$, so we can define W^P to represent the width of walkways between every two machine columns. Note that the front side of the machines are required to be lined-up on one sides of a walkway. Thus, $P^{\max}$ is the upper bound of the required number of walkways, and the space consumption of a walkway does not include in the required space if there is not any machines on one side of the walkway.

For defining decision variables, let w^R_r represent the occupied width for row r, where $r \in \mathcal{R}$, and let $w^F \geq max\{w^R_r,\ \forall\ r \in \mathcal{R}\}$ represent the total occupied width of the floor. Then, we define $x^M_{m,f}$ and $y^M_{m,f}$ as the lower left coordinates of the position of machine m from functional area f, where $m \in \mathcal{M}_f$, and $f \in \mathcal{F}$. Then we set $x^P_{p,r}$ as the left edge x-coordinate of walkway p in row r, where $p \in \mathcal{P}$ and $r \in \mathcal{R}$. In order to prevent overlapping in the layout plans, we apply binary variables to determine the relative locations of two objects, such as two machines, one machine and a walkway, or two walkways. Let $b^{M,\text{left}}_{m,n} = 1$ ($b^{M,\text{below}}_{m,n} = 1$) if physical machine m is placed to the left of machine n (or below) machine n, where m, $n \in \mathcal{M}_f$, $m \neq n$, and $f \in \mathcal{F}$, and let $b^{M,P}_{m,p,r} = 1$ ($b^{P,M}_{m,p,r} = 1$) if physical machine m is placed to the left (right) of walkway p in row r, where $m \in \mathcal{M}_f$, $f \in \mathcal{F}$, $p \in \mathcal{P}$, and $r \in \mathcal{R}$. Then we define $b^{P,\text{left}}_{p,q,r} = 1$ if physical walkway p is placed to the left of walkway q, where p, $q \in \mathcal{P}$, $p \neq q$, and $r \in \mathcal{R}$. Furthermore, as described, the problem involves machine assignment, which is to determine which machines from a FA should be placed on which side of a walkway in which row. Hence, let $a^M_{m,r} = 1$ if machine m from functional area f is assigned (placed) in the row r, where $m \in \mathcal{M}_f$, $f \in \mathcal{F}$, and $r \in \mathcal{R}$, and let $a^P_{p,r} = 1$ if at least one machine is assigned (placed) on the either edge of walkway p in row r (walkway p is needed in row r), where $p \in \mathcal{P}$, and $r \in \mathcal{R}$. For determining the relative location of FAs, we define $b^{S,\text{left(right)}}_{m,f,p,r} = 1$ if physical machine m from the functional area f is placed on the left (right) side edge of walkway p in row r , where $m \in \mathcal{M}_f$, $f \in \mathcal{F}$, $p \in \mathcal{P}$, and $r \in \mathcal{R}$.

Finally, the formulations for the lined-up requirement of machines are non-linear, and we linearize the constraints by introducing $l^{S,\text{left(right)}}_{m,f,p,r}$, which represents $b^{S,\text{left(right)}}_{m,f,p,r} x^P_{p,r}$ for linearizing Constraint (2.20), where $m \in \mathcal{M}_f$, $f \in \mathcal{F}$, $p \in \mathcal{P}$, and $r \in \mathcal{R}$. Details of the constraints will be introduced later. A summary of all parameters and decision variable employed by the MILP model are provided in Tables 2.2 and 2.3, respectively, while the size notations are illustrated in Figure 2.3 for ease of understanding. The remainder of this section is devoted to the mixed integer linear programming (MILP) formulation for the proposed problem.

Table 2.2: Summary of parameter notations

$\mathcal{F}$	Set of functional areas.
$\mathcal{M}_f$	Set of machines from functional area f, where $f \in \mathcal{F}$.
$\mathcal{P}$	Set of possible walkways in each row, where $\mathcal{P} = \{1, 2, \ldots, P^{\max}\}$.
$\mathcal{R}$	Set of rows, where $\mathcal{R} = \{1, 2, \ldots\}$.
$\Phi_{f,r}$	The order of functional area f in row r from left to right. For example, if functional area f is the third functional area from the left in row r, then $\Phi_{f,r} = 3$. Note if functional area f does not appear in row r then $\Phi_{f,r} = 0$, where $f \in \mathcal{F}$ and $r \in \mathcal{R}$.
$C^{M,B}_{m,n}$	The required back clearance between machine m and machine n. $C^{M,b}_{m,o}$ represents the back clearance requirement of machine m (also the clearance requirement between machine m and the wall), so $C^{M,B}_{m,n}$ =max$\{C^{M,B}_{m,o}, C^{M,B}_{n,o}\}$, where $m, n \in \mathcal{M}_f$, $f \in \mathcal{F}$, and $m \neq n$.
$C^{M,S}_{m,n}$	The required side clearance between machine m and machine n, where $m, n \in \mathcal{M}_f$, $f \in \mathcal{F}$, and $m \neq n$. As $C^{M,b}_{m,o}$, $C^{M,S}_{m,o}$ represents the back clearance requirement of machine m.
H^M_m	The height of machine i, where $m \in \mathcal{M}_f$, and $f \in \mathcal{F}$.
H^R_r	The available height of row r, where $r \in \mathcal{R}$.
W^F	The width of the floor.
W^M_m	The width of machine m, where $m \in \mathcal{M}_f$, and $f \in \mathcal{F}$.
W^P	The width of walkways between every two machine columns.
Y^L_r	The lower edge y-coordinates of row r, where $r \in \mathcal{R}$.
Y^U_r	The upper edge y-coordinates of row r, where $r \in \mathcal{R}$.

This problem seeks to determine whether a given collection of machines could be feasibly located within the existing facility. To answer this question, the objective of the MILP

Table 2.3: Summary of decision variable notations

$a^{M}_{m,f,r} \in \{0,1\}$	$a^{M}_{m,r} = 1$ if machine m from functional area f is assigned (placed) in the row r, where $m \in \mathcal{M}_f$, $f \in \mathcal{F}$, and $r \in \mathcal{R}$.
$a^{P}_{p,r} \in \{0,1\}$	$a^{P}_{p,r} = 1$ if at least one machine is assigned (placed) on the either edge of walkway p in row r (walkway p is needed in row r), where $p \in \mathcal{P}$, and $r \in \mathcal{R}$.
$b^{S,\text{left(right)}}_{m,f,p,r} \in \{0,1\}$	$b^{S,\text{left(right)}}_{m,f,p,r} = 1$ if physical machine m from the functional area f is placed on the left(right) side edge of walkway p in row r , where $m \in \mathcal{M}_f$, $f \in \mathcal{F}$, $p \in \mathcal{P}$, and $r \in \mathcal{R}$.
$b^{M,\text{below}}_{m,n} \in \{0,1\}$	$b^{M,\text{below}}_{m,n} = 1$ if physical machine m is placed below machine n, where m, $n \in \mathcal{M}_f$, $m \neq n$, and $f \in \mathcal{F}$.
$b^{M,\text{left}}_{m,n} \in \{0,1\}$	$b^{M,\text{left}}_{m,n} = 1$ if physical machine m is placed to the left of machine n, where m, $n \in \mathcal{M}_f$, $m \neq n$, and $f \in \mathcal{F}$.
$b^{M,P}_{m,p,r} \in \{0,1\}$	$b^{M,P}_{m,p,r} = 1$ if physical machine m is placed to the left of walkway p in row r, where $m \in \mathcal{M}_f$, $f \in \mathcal{F}$, $p \in \mathcal{P}$, and $r \in \mathcal{R}$.
$b^{P,M}_{m,p,r} \in \{0,1\}$	$b^{P,M}_{m,p,r} = 1$ if physical machine m is placed to the right of walkway p in row r. Note that $b^{P,M}_{m,p,r} + b^{P,M}_{m,p,r} = 0$ if machine m is not assigned in row r, where $m \in \mathcal{M}_f$, $f \in \mathcal{F}$, $p \in \mathcal{P}$, and $r \in \mathcal{R}$.
$b^{P,\text{left}}_{p,q,r} \in \{0,1\}$	$b^{P,\text{left}}_{p,q,r} = 1$ if physical walkway p is placed to the left of walkway q, where p, $q \in \mathcal{P}$, $p \neq q$, and $r \in \mathcal{R}$.
$l^{S,\text{left(right)}}_{m,f,p,r}$	To represent $b^{S,\text{left(right)}}_{m,f,p,r} x^{P}_{p,r}$ for linearizing Constraint (2.20), where $m \in \mathcal{M}_f$, $f \in \mathcal{F}$, $p \in \mathcal{P}$, and $r \in \mathcal{R}$.
w^F	Total occupied width of the floor; $w^F \geq max\{w^R_r,\ \forall\ r \in \mathcal{R}\}$.
w^R_r	The occupied width for row r, where $r \in \mathcal{R}$.
$x^{M}_{m,f}, y^{M}_{m,f}$	The lower left coordinates of the position of machine m from functional area f, where $m \in \mathcal{M}_f$, and $f \in \mathcal{F}$.
$x^{P}_{p,r}$	The left edge x-coordinate of walkway p in row r, where $p \in \mathcal{P}$ and $r \in \mathcal{R}$.

formulated below is to minimize the total required width of the facility, w^F, given that the total height dimension is fixed (i.e., the heights of each row, H^R_r, and the walkways between rows are known). Thus, any solution with an objective function value less than the width of the current facility would indicate that this facility is sufficiently large.

The MILP model of the MRMIC layout design problem is provided below.

$$\text{Min} \quad w^F \tag{2.1}$$

$$\text{s.t.} \quad w^F \geq w^R_r \quad \forall\ r \in \mathcal{R}, \tag{2.2}$$

$$x^{M}_{m,f} + W^{M}_{m} + C^{M,B}_{m,o} \leq w^R_r, \quad \forall\ m \in \mathcal{M}_f, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.3}$$

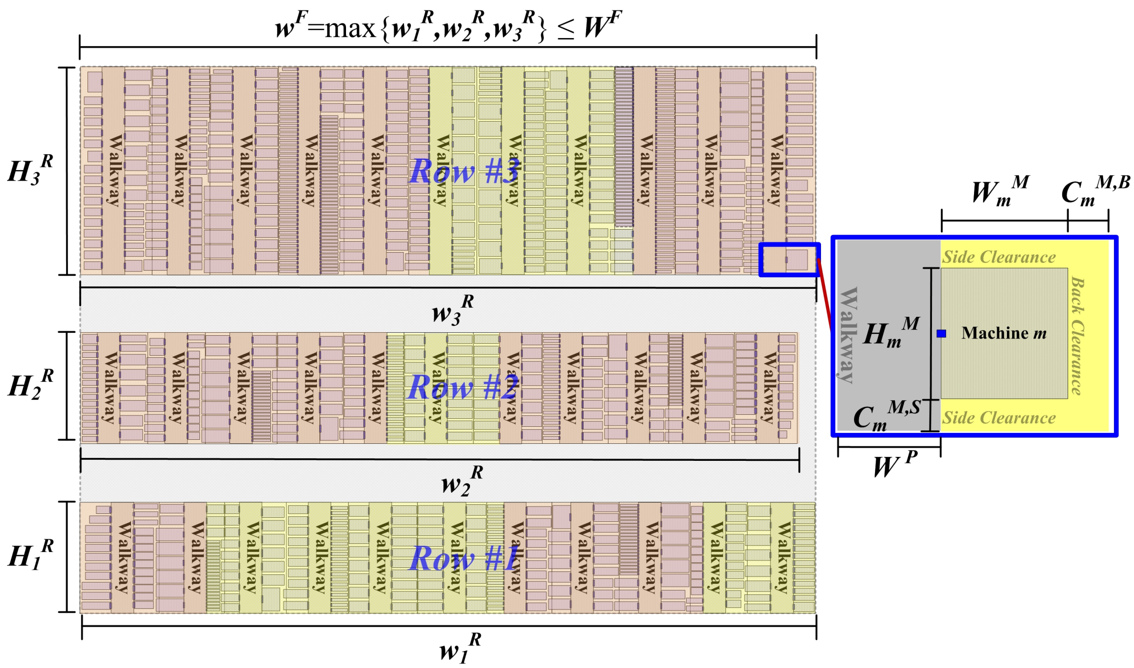


Figure 2.3: An illustration of the facility dimension notation employed in the MRMIC model

$$x^P_{p,r} + W^P \leq w^R_r + W^F(1 - a^P_{p,r}), \quad \forall\, p \in \mathcal{P}, r \in \mathcal{R}, \tag{2.4}$$

$$\sum_{m \in \mathcal{M}_f} (b^{S,\text{left}}_{m,f,p,r} + b^{S,\text{right}}_{m,f,p,r}) \leq P^{\max} a^P_{p,r}, \quad \forall\, f \in \mathcal{F}, p \in \mathcal{P}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.5}$$

$$b^{M,\text{left}}_{m,n} + b^{M,\text{left}}_{n,m} + b^{M,\text{below}}_{m,n} + b^{M,\text{below}}_{n,m} \geq 1, \quad \forall\, m, n \in \mathcal{M}_f, m \neq n, f \in \mathcal{F}, \tag{2.6}$$

$$b^{P,\text{left}}_{p,q,r} + b^{P,\text{left}}_{q,p,r} = 1, \quad \forall\, p, q \in \mathcal{P}, p \neq q, r \in \mathcal{R}, \tag{2.7}$$

$$\sum_{r \in \mathcal{R}} a^M_{m,f,r} = 1, \quad \forall\, m \in \mathcal{M}_f, f \in \{\mathcal{F} : \Phi_{f,r} > 0\}, \tag{2.8}$$

$$b^{M,P}_{m,p,r} + b^{P,M}_{m,p,r} = a^M_{m,f,r}, \quad \forall\, m \in \mathcal{M}_f, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.9}$$

$$x^M_{m,f} - x^M_{n,f} + W^F b^{M,\text{left}}_{m,n} \leq W^F - W^M_m - C^{M,B}_{m,n},$$
$$\forall\, m, n \in \mathcal{M}_f, m \neq n, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.10}$$

$$y^M_{m,f} - y^M_{n,f} + H^R_r b^{M,\text{below}}_{m,n} \leq H^R_r - H^M_m - C^{M,S}_{m,n},$$
$$\forall\, m, n \in \mathcal{M}_f, m \neq n, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.11}$$

$$x^M_{m,r} - x^P_{p,r} + W^F b^{M,P}_{m,p,r} \leq W^F - W^M_m,$$
$$\forall\, m \in \mathcal{M}_f, p \in \mathcal{P}, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.12}$$

$$x^P_{p,r} - x^M_{m,r} + W^F b^{P,M}_{m,p,r} \le W^F - W^P, \quad \forall\, m \in \mathcal{M}_f, p \in \mathcal{P}, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.13}$$

$$x^P_{p,r} - x^P_{q,r} + W^F b^{P,\text{left}}_{p,q,r} \le W^F - W^P, \quad \forall\, p, q \in \mathcal{P}, p \ne q, r \in \mathcal{R}, \tag{2.14}$$

$$y^M_{m,f} \le \sum_{r\in\mathcal{R}} Y^U_r (1 - a^M_{m,f,r}) - C^{M,S}_{m,o}, \quad \forall\, m \in \mathcal{M}_f, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.15}$$

$$x^M_{m,f} \ge C^{M,B}_{m,o}, \quad \forall\, m \in \mathcal{M}_f, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.16}$$

$$y^M_{m,f} \ge C^{M,S}_{m,o} + \sum_{r\in\mathcal{R}} Y^L_r (1 - a^M_{m,f,r}), \quad \forall\, m \in \mathcal{M}_f, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.17}$$

$$\sum_{m\in\mathcal{M}_f} \sum_{p\in\mathcal{P}} (b^{S,\text{left}}_{m,f,p,r} + b^{S,\text{right}}_{m,f,p,r}) \ge 1, \quad \forall\, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.18}$$

$$p(b^{S,\text{left}}_{m,f,p,r} + b^{S,\text{right}}_{m,f,p,r}) - p(b^{S,\text{left}}_{m',f',r,p} + b^{S,\text{right}}_{m',f',r,p}) \le 0,$$
$$\forall\, m \in \mathcal{M}_f, m' \in \mathcal{M}'_f, f, f' \in F, p \in \mathcal{P}, r \in \{\mathcal{R} : \Phi_{f',r} > \Phi_{f,r} > 0, \}, \tag{2.19}$$

$$\sum_{p\in\mathcal{P}} [b^{S,\text{left}}_{m,f,p,r}(x^P_{p,r} - W^M_m) + b^{S,\text{right}}_{m,f,p,r}(x^P_{p,r} + W^P)] = x^M_{m,f},$$
$$\forall\, m \in \mathcal{M}_f, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.20}$$

$$a^M_{m,f,r}, a^P_{p,r}, b^{P,\text{left}}_{p,q,r}, b^{S,\text{left}}_{m,f,p,r}, b^{S,\text{right}}_{m,f,p,r}, b^{M,P}_{m,p,r}, b^{P,M}_{m,p,r}, \in \{0, 1\},$$
$$\forall\, m \in \mathcal{M}_f, p, q \in \mathcal{P}, p \ne q, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.21}$$

$$b^{M,\text{below}}_{m,n}, b^{M,\text{left}}_{m,n} \in \{0, 1\}, \quad \forall\, m, n \in \mathcal{M}_f, m \ne n, \tag{2.22}$$

$$x^P_{p,r}, x^M_{m,f}, y^M_{m,f}, w^F, w^R_r \ge 0, \quad \forall\, m \in \mathcal{M}_f, p \in \mathcal{P}, f \in \mathcal{F}, r \in \mathcal{R}. \tag{2.23}$$

The objective function (2.1) and Constraint (2.2), which determines the maximum occupied width of any row, serve to minimize the total required floor area in terms of minimizing the maximum used row width. Constraints (2.3) and (2.4) determine the required floor width for the layout, where the widths of unused walkways (e.g., where no machines are located on either side) are ignored. Constraint (2.5) determines which walkways are not used. Constraints (2.6)–(2.9) require every machine and walkway to be placed on either side edge of a walkway in exactly one row. Constraints (2.10)–(2.14) prevent overlapping machines and walkways, while Constraints (2.15)–(2.17) ensure that all machines and walkways are located within the facility boundaries. Constraint (2.18) ensures at least one machine from functional area f is placed in the designated row $r \in \{\mathcal{R} : \Phi_{f,r} > 0\}$ in the given relative location. Constraint (2.19) ensures that all machines are not violating the given relative location of FAs in each row. Constraint (2.20), which is the only non-linear constraint in this model, ensures that all machines are aligned on either side edge of a walkway. Finally, Constraints (2.21)–(2.23) describe the decision variable definitions.

By introducing new binary decision variables $l^{S,\text{left}}_{m,f,p,r}$ and $l^{S,\text{right}}_{m,f,p,r}$, Constraint (2.20) can be linearized as follows:

$$l^{S,\text{left}}_{m,f,p,r} \leq W^F b^{S,\text{left}}_{m,f,p,r}, \quad \forall\, m \in \mathcal{M}_f, p \in \mathcal{P}, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.24}$$

$$l^{S,\text{right}}_{m,f,p,r} \leq W^F b^{S,\text{right}}_{m,f,p,r}, \quad \forall\, m \in \mathcal{M}_f, p \in \mathcal{P}, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.25}$$

$$l^{S,\text{left}}_{m,f,p,r} \geq x^P_{p,r} - W^F(1 - b^{S,\text{left}}_{m,f,p,r}), \quad \forall\, m \in \mathcal{M}_f, p \in \mathcal{P}, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.26}$$

$$l^{S,\text{right}}_{m,f,p,r} \geq x^P_{p,r} - W^F(1 - b^{S,\text{right}}_{m,f,p,r}) \quad \forall\, m \in \mathcal{M}_f, p \in \mathcal{P}, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.27}$$

$$l^{S,\text{left}}_{m,f,p,r} \leq x^P_{p,r}, \quad \forall\, m \in \mathcal{M}_f, p \in \mathcal{P}, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.28}$$

$$l^{S,\text{right}}_{m,f,p,r} \leq x^P_{p,r}, \quad \forall\, m \in \mathcal{M}_f, p \in \mathcal{P}, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.29}$$

$$\sum_{p \in \mathcal{P}} (l^{S,\text{left}}_{m,f,p,r} - W^M_m b^{S,\text{left}}_{m,f,p,r} + l^{S,\text{right}}_{m,f,p,r} + W^P b^{S,\text{right}}_{m,f,p,r}) = x^M_{m,f},$$
$$\forall\, m \in \mathcal{M}_f, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}, \tag{2.30}$$

$$l^{S,\text{left}}_{m,f,p,r}, l^{S,\text{right}}_{m,f,p,r} \geq 0, \quad \forall\, m \in \mathcal{M}_f, p \in \mathcal{P}, f \in \mathcal{F}, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}. \tag{2.31}$$

As mentioned, the MRMIC problem is an amalgamation of multiple NP-hard problems, which makes small-scale problems difficult, and practically-sized large-scale problems impossible, to solve directly. To overcome this issue, a heuristic solution approach that utilizes two integer programming sub-problems is proposed.

2.4 Solution approach

The MRMIC problem seeks to determine the exact location of each machine to minimize the total required space. The problem incorporates both continuous (locations of machines) and combinatorial (machine assignment and relative locations of machines) decision variables. Preliminary testing indicated that MILP solvers may require hours to solve a simple 10-machine problem. Thus, a construction heuristic is proposed to reduce the problems to a practical size to enable them to be solved by starting with a manufacturing facility that is initially empty.

As introduced in Section 2.4.1, the heuristic begins by assigning a priority value to all unallocated FAs, based on a continuous-space approximation of the space required by this FA. Once the order is determined, an iterative procedure is used to arrange the occupied space for the functional area with the highest priority. This is repeated until all FAs are allocated. Second, as described in Section 2.4.2, a linear programming (LP) model is solved to estimate the amount of space required for the entire layout, based on the continuous-space relaxation of the area required for each FA. This provides guidance to allocate machines to a specific row in the case where a single FA appears in multiple rows. The estimation includes the space occupied by the allocated FAs and continuous relaxation of FAs that are currently un-allocated, both of which are considered as parameters for the following step. Next, as presented in Section 2.4.3, two integer programs are solved to determine the tool layout for each FA. Note that FA interlocking is not considered in the sub-problems, but is instead considered in the procedure described in Section 2.4.4, where FA interlocking is performed for I. Then, as introduced in Section 2.4.5, a procedure incorporating machine interlocking is applied for better space utilization. The above procedures solve the MRMIC when the relative location of FAs is pre-determined because, in practice, some semiconductor manufacturers may have their own production process and their own way to manage contamination. Therefore, for the situation that the relative location of FAs is flexible, we propose a neighborhood search procedure in Section 2.4.6 to determine whether the solution could be improved by relaxing the given relative location. Note that the proposed heuristic can be used to solve the MRMIC under either fixed or flexible relative locations of FAs by incorporating the neighborhood search procedure.

Pseudocode of the proposed heuristic is provided in Algorithm 1.

Algorithm 1 Main procedure

Initialize $\mathcal{U}$; % Section 2.4.1
$\mathcal{A} = \{\varnothing\}$
for all $u \in \mathcal{U}$ **do**
  $\mathcal{A} \leftarrow \mathcal{A} \cup u$; % $\mathcal{A}$ represents the list of FAs to be allocated
  $[w^{E}_{f'',r}]$=`L1`$(A^{O}_{f',r})$; % lines 2-5 in Section 2.4.2
  $[x^{G}_{g,k,r}, w^{K,\text{upper}}_{k,r}]$=`S1`$(A^{O}_{f',r}, H^{K,\text{left}}_{u,r}, H^{K,\text{right}}_{u,r}, w^{E}_{f'',r}, W^{K,\text{left}}_{u,r}$, and $W^{K,\text{right}}_{u,r})$
  if At least one of the adjacent FAs has not been allocated **then**
    $[x^{G}_{g,k,r}]$=`S2`$(h^{K}_{k^{\text{B}},r}, x^{G}_{g,k,r}, w^{K,\text{upper}}_{k,r})$
  end if
    % Lines 6-10 in Section 2.4.3
  $[Z^{\phi}_{r}]$=`FAI`$(u, x^{G}_{g,k,r}, Z^{\phi}_{r})$ % Section 2.4.4
  $[A^{O}_{f',r}, H^{K,\text{left}}_{f,r}, H^{K,\text{right}}_{f,r}, W^{K,\text{left}}_{f,r}, W^{K,\text{left}}_{f,r}, Z^{\phi}_{r}]$=`PA`$(H^{K,\text{left}}_{f,r}, H^{K,\text{right}}_{f,r}, u, W^{K,\text{left}}_{f,r}, W^{K,\text{left}}_{f,r}, Z^{\phi}_{r})$
    % Section 2.4.5
end for
$[A^{O}_{f',r}, Z^{\phi}_{r}]$=`NS`$(A^{O}_{f',r}, H^{K,\text{left}}_{f,r}, H^{K,\text{right}}_{f,r}, \mathcal{U}, W^{K,\text{left}}_{f,r}, W^{K,\text{left}}_{f,r}, Z^{\phi}_{r})$ % Section 2.4.6

2.4.1 Determine the priority order of all un-allocated functional areas

Recall that the MRMIC model is designed to minimize the total amount of required space in terms of minimizing the maximum width of all rows. Further, recall that some FAs may appear in multiple rows (as determined *a priori*). To avoid over-allocating machines to rows that already have long width requirements, FAs that appear in more than one row are assigned a lower priority. The size of the continuous space approximation is used to break ties (a smaller size corresponds to a higher priority). The motivation behind this priority-ranking scheme is that FAs that appear in multiple rows have greater flexibility and should thus be assigned later in the process. The continuous approximation of required space for functional area f, denoted as $\mathcal{B}^{F}_{f}$, is calculated as

$$\mathcal{B}^F_f = \sum_{\forall m \in \mathcal{M}_f} (H^M_m + C^{M,S}_m)[W^M_m + 0.5(C^{M,B}_m + W^P)], \ \forall \ f \in \mathcal{F}.$$

This expression determines a minimum estimate of the required amount of space based on the sum of machine space, one side clearance, and one-half back clearance. In addition, the estimate also includes space for a connected walkway based on the sum of the machine space, one side clearance, one-half back clearance, and the space requirement for the connected walkway. An example for determining the priority order is provided below. Suppose the relative locations of FAs for a two row MRMIC problem with three FAs is such that Films $\rightarrow$ Photolithography for the first row and Diffusion $\rightarrow$ Films for the second row. Given $\mathcal{B}^F_{\text{Diffusion}} < \mathcal{B}^F_{\text{Photolithography}}$, the priority order is determined as $\mathcal{U} = \{\text{Diffusion}, \text{Photolithography}, \text{Films}\}$, which means Films will be the last one to be allocated since it appears twice in the given relative locations. Diffusion will be allocated before Photolithography since $\mathcal{B}^F_{\text{Diffusion}} < \mathcal{B}^F_{\text{Photolithography}}$.

2.4.2 Estimate the space consumption of un-allocated FAs

Once a FA has been selected for insertion into the layout, the next step of the heuristic is to estimate the space that will be consumed by the remaining un-allocated FAs. This estimate will be used to guide the the machine allocation of FA u in which rows if u appears in multiple rows in the next stage of the heuristic. Obviously it is impossible to know the exact area that will be consumed by these functional areas at this stage in the heuristic, as the allocation of machines to particular locations is performed later. Instead, the following LP model, named "Model `L1`", is utilized to provide the estimate. This LP model requires some new notations. Specifically, let $A^O_{f',r}$ represent the occupied space of the allocated FA $f' \in \mathcal{A} \backslash u$ in row r, and decision variable $w^E_{f'',r}$ represent the estimated required width of un-allocated FA $f'' \in (\mathcal{F}\backslash\mathcal{A}) \cup u$ in row r.

$$\text{(Model L1): Min} \quad w^F \tag{2.32}$$

$$\text{s.t.} \quad \sum_{f' \in \{\mathcal{A}\backslash u : \Phi_{f',r} > 0\}} A^O_{f',r}/H^R_r + \sum_{f'' \in \{(\mathcal{F}\backslash\mathcal{A}) \cup u : \Phi_{f'',r} > 0\}} w^E_{f'',r} \le w^F,$$

$$\forall\ r \in \mathcal{R}, \tag{2.33}$$

$$\sum_{r \in \mathcal{R}} H^R_r w^E_{f'',r} = \mathcal{B}^F_{f''}, \quad \forall\ f'' \in \{(\mathcal{F}\backslash\mathcal{A}) \cup u : \Phi_{f'',r} > 0\}, \tag{2.34}$$

$$H^R_r w^E_{f'',r} \ge \min_{\forall m \in \mathcal{M}_{f''}} \{(H^M_m + C^{M,S}_m)[W^M_m + 0.5(C^{M,B}_m + W^P)]\},$$

$$\forall\ f'' \in \{(\mathcal{F}\backslash\mathcal{A}) \cup u : \Phi_{f'',r} > 0\}, r \in \mathcal{R}, \tag{2.35}$$

$$w^E_{f'',r}, w^F \ge 0, \quad \forall\ f'' \in (\mathcal{F}\backslash\mathcal{A}) \cup u, r \in \{\mathcal{R} : \Phi_{f'',r} > 0\}. \tag{2.36}$$

The objective function (2.32) seeks to minimize the required floor width. Corresponding to the objective function, Constraint (2.33) determines the maximum occupied width of all rows. Constraint (2.34) determines the estimated required width of un-allocated FAs based on the continuous approximation. Constraint (2.35) ensures the estimated width of FA f'' in row r at least equal to the size of the smallest machine from f'' if f'' appears in row r. Constraint (2.36) defines the decision variables. After solving Model L1, we now use the values of $w^E_{f'',r}$ to estimate the required width of un-allocated FAs while allocating machines from u.

2.4.3 Establish the machine allocation within FA u

Due to the NP-hardness of solving the MRMIC problem, it is impractical to solve the problem with even only one FA directly by integer programming solvers. Even when isolating a single FA, it is still impractical to employ integer programming solvers to determine the exact tool locations for this FA. One way to simplify the problem is to ignore the tool interlocking possibility. Thus, assuming machine interlocking is not allowed, with only one FA, the MRMIC problem can be considered as a collection of sub-problems, which assign machines to independent machine columns on the floor. Therefore, Models `S1` and `S2` are proposed for solving the sub-problems with the notations described and illustrated in Tables 2.4 and 2.5, and Figure 2.4, respectively.

`S1` presents the layout of u with an objective to minimize total required floor width, but may result in alternative solutions with shorter columns. These alternative solutions have less available space on boundary columns, the columns next to or shared with the adjacent FAs, for adjacent un-arranged FAs within the same rows. Therefore, `S2` is presented to minimize the height of the boundary columns for u after solving `S1`. Note that, because of the non-linearity of minimized required space by the multiplication of occupied widths and heights for machine columns, `S1` and `S2` are solved sequentially.

Parameters of the sub-problems are introduced. In order to reduce computational efforts, we categorize same size machines with the same clearance requirements from u as a group, $\mathcal{G}$. Moreover, we define H_g^G as the height of machines from group g, N_g^G as the number of machines from equipment group g, and W_g^G as the width of machines from group g, where $g \in \mathcal{G}$. Additionally, let $\mathcal{K}_r$ represents the set of machine columns in row r, where $\mathcal{K}_r = \{1, 2, \ldots, |\mathcal{K}_r|\}$, and $|\mathcal{K}_r| \leq [\sum_g (C_g^{G,S} + H_g^G)]/(H_r^R - \max\{C_g^{G,S},\ \forall\ g \in \mathcal{G}\})$, where $g \in \mathcal{G}$, $r \in \mathcal{R}$. Note that $\mathcal{K}_r = 0$ if $\Phi_{u,r} = 0$. Besides, let $C_g^{G,B}$ ($C_g^{G,S}$) represent the required back (side) clearance of machines from group g, where $g \in \mathcal{G}$. Furthermore, let $G_{g,k,r}^{\max}$ represents the upper bound of the machine amount from group g in column k of row r. $G_{g,k,r}^{\max}$ =min$\{N_g^G, H_r^R/(H_g^G + C_g^{G,S})\}$, where $g \in \mathcal{G}$, $k \in \mathcal{K}_r$, and $r \in \{\mathcal{R} : \Phi_{u,r} > 0\}$. We will introduce how to use the upper bound later in introducing variables of the sub-problems.

For variables in the sub-problems, let $x_{g,k,r}^G$ represent the number of machines from the group g assigned to column k of row r, and $x_{g,k,r}^G \leq G_{g,k,r}^{\max}$, where $g \in \mathcal{G}$, $k \in \mathcal{K}_r$, and $r \in \{\mathcal{R} : \Phi_{u,r} > 0\}$. In order to determine the size of a machine column in the sub-problems, we set $h_{k,r}^K$ ($w_{k,r}^K$) as the occupied height (width) of the column k in row r, where $k \in \mathcal{K}_r$, and $r \in \{\mathcal{R} : \Phi_{u,r} > 0\}$. And also, as introduced, `S1` and `S2` are solved sequentially, so we have $w_{k,r}^{K,\text{upper}}$ as the used width upper bound of column k in row r solved by Model `S1` and applied as a constant in Model S2, where $k \in \mathcal{K}_r$, and $r \in \{\mathcal{R} : \Phi_{u,r} > 0\}$. Next, let $c_{k,r}^{K,\max}$ represent the maximum clearance of all machines in column k of row r, where $k \in \mathcal{K}_r$, and $r \in \{\mathcal{R} : \Phi_{u,r} > 0\}$. In the sub-problems, if a property of total side clearance requirements between machines in a column for a set of machines is provided, S1 and S2 determine only the number of machines from which group in which row. Thus, the property is proved later in this section, which demonstrates that the minimum total side clearance in a column is the sum of the $C_g^{G,S}$ for all machines in the column and the maximum one side clearance $c_{k,r}^{K,\max}$. Hence, we set $b_{g,k,r} = 1$ if machines from the group g have the greatest clearance requirement in column k of row r, where $g \in \mathcal{G}$, $k \in \mathcal{K}_r$, and $r \in \{\mathcal{R} : \Phi_{u,r} > 0\}$. Finally, we set w_r^P to represent the required amount of walkways in row r, $r \in \{\mathcal{R} : \Phi_{u,r} > 0\}$.

While the heuristic progresses, if adjacent FAs of u have been allocated, the size information of the boundary columns in the adjacent FAs is set as bounds for adjacent columns of u when solving the sub-problems for functional area interlocking. Let $H_{u,r}^{K,\text{left}}$ ($H_{u,r}^{K,\text{right}}$) and $W_{u,r}^{K,\text{left}}$ ($W_{u,r}^{K,\text{right}}$) be the occupied heights and widths of boundary columns of the adjacent allocated FA on the left (right) side of u in row r, respectively. These heights and widths are determined in Section 2.4.5 once a functional area has been allocated. For example, if the boundary column on the right side of the adjacent FA in row r is just half-used, then the first column of u in row r is bounded by the other half available space with a height as $(H_r^R - H_{u,r}^{K,\text{left}})$ and width as $W_{u,r}^{K,\text{left}}$ when solving the sub-problems for functional area interlocking.

S1 is described below:

$$\text{(Model S1): Min} \quad w^F \tag{2.37}$$

$$\text{s.t.} \quad \sum_{f' \in \mathcal{A} \setminus u, \Phi_{f',r} > 0} A_{f',r}^O / H_r^R + \sum_{f'' \in (\mathcal{F} \setminus \mathcal{A}), \Phi_{f'',r} > 0} w_{f'',r}^E + \sum_{k \in \mathcal{K}_r} w_{k,r}^K + W^P w_r^P \leq w^F,$$

Table 2.4: Parameters of the sub-problems

$\mathcal{G}$	Set of equipment groups from functional area u, where $\mathcal{G} = \{1, 2, \ldots\}$.
$\mathcal{K}_r$	Set of columns in row r, where $\mathcal{K}_r = \{1, 2, \ldots, \lvert\mathcal{K}_r\rvert\}$, and $\lvert\mathcal{K}_r\rvert \leq [\sum_g (C_g^{G,S} + H_g^G)]/(H_r^R - \max\{C_g^{G,S},\ \forall\ g \in \mathcal{G}\})$, where $g \in \mathcal{G}$, $r \in \mathcal{R}$. Note that $\mathcal{K}_r = 0$ if $\Phi_{u,r} = 0$.
$C_g^{G,B}$	The required clearance on back sides of machines from group g, where $g \in \mathcal{G}$.
$C_g^{G,S}$	The required side clearance of machines from group g, where $g \in \mathcal{G}$.
$G_{g,k,r}^{\max}$	The upper bound of the machine amount from group g in column k of row r. $G_{g,k,r}^{\max} = \min\{N_g^G, H_r^R/(H_g^G + C_g^{G,S})\}$, where $g \in \mathcal{G}$, $k \in \mathcal{K}_r$, and $r \in \{\mathcal{R} : \Phi_{u,r} > 0\}$.
H_g^G	The height of machines from group g, where $g \in \mathcal{G}$.
N_g^G	Number of machines from equipment group g, where $g \in \mathcal{G}$.
W_g^G	The width of machines from group g, where $g \in \mathcal{G}$.

Table 2.5: Decision variables of the sub-problems

$b_{g,k,r}^K \in \{0, 1\}$	$b_{g,k,r} = 1$ if machines from the group g have the greatest clearance requirement in column k of row r, where $g \in \mathcal{G}$, $k \in \mathcal{K}_r$, and $r \in \{\mathcal{R} : \Phi_{u,r} > 0\}$.
$c_{k,r}^{K,\max}$	The maximum clearance of all machines in column k of row r, where $k \in \mathcal{K}_r$, and $r \in \{\mathcal{R} : \Phi_{u,r} > 0\}$.
$h_{k,r}^K$	The occupied height of the column k in row r, where $k \in \mathcal{K}_r$, and $r \in \{\mathcal{R} : \Phi_{u,r} > 0\}$.
$x_{g,k,r}^G$	The amount of machines from the group g is assigned into column k of row r, and $x_{g,k,r}^G \leq G_{g,k,r}^{\max}$, where $g \in \mathcal{G}$, $k \in \mathcal{K}_r$, and $r \in \{\mathcal{R} : \Phi_{u,r} > 0\}$.
$w_{k,r}^K$	The used width of column k in row r, where $k \in \mathcal{K}_r$, and $r \in \{\mathcal{R} : \Phi_{u,r} > 0\}$.
$w_{k,r}^{K,\text{upper}}$	The used width upper bound of column k in row r for Model `S2` (solved by Model `S1`), where $k \in \mathcal{K}_r$, and $r \in \{\mathcal{R} : \Phi_{u,r} > 0\}$.
w_r^P	The required amount of walkways in row r, $r \in \{\mathcal{R} : \Phi_{u,r} > 0\}$.

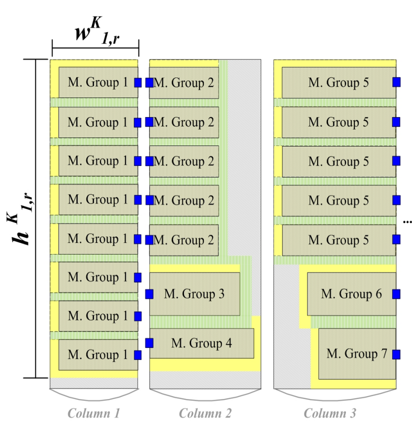


Figure 2.4: Illustration of size notations in the sub-problems

$$\forall r \in \{\mathcal{R} : \Phi_{u,r} > 0\}, \tag{2.38}$$

$$\sum_{r \in \{\mathcal{R}:\Phi_{u,r}>0\}} \sum_{k \in \mathcal{K}_r} x^G_{g,k,r} = N^G_g, \quad \forall\ g \in \mathcal{G}, \tag{2.39}$$

$$G^{\max}_{g,k,r} b^K_{g,k,r} \geq x^G_{g,k,r}, \quad \forall\ g \in \mathcal{G}, k \in \mathcal{K}_r, r \in \{\mathcal{R} : \Phi_{u,r} > 0\}, \tag{2.40}$$

$$C^{G,S}_g b^K_{g,k,r} \leq c^{K,\max}_{k,r}, \quad \forall\ g \in \mathcal{G}, k \in \mathcal{K}_r, r \in \{\mathcal{R} : \Phi_{u,r} > 0\}, \tag{2.41}$$

$$\sum_{g \in \mathcal{G}} (H^G_g + C^{G,S}_g) x^G_{g,k,r} + c^{K,\max}_{k,r} \leq h^K_{k,r},$$

$$\forall\ k \in \mathcal{K}_r, r \in \{\mathcal{R} : \Phi_{u,r} > 0\}, \tag{2.42}$$

$$h^K_{k,r} \leq H^R_r, \quad \forall\ k \in \mathcal{K}_r, r \in \{\mathcal{R} : \Phi_{u,r} > 0\}, \tag{2.43}$$

$$(W^G_g + C^{G,B}_g) b^G_{g,k,r} \leq w^K_{k,r},$$

$$\forall\ g \in \mathcal{G}, k \in \mathcal{K}_r, r \in \{\mathcal{R} : \Phi_{u,r} > 0\}, \tag{2.44}$$

$$\sum_{g \in \mathcal{G}} \sum_{k \in \mathcal{K}_r} x^G_{g,k,r} \geq 1, \quad \forall\ r \in \{\mathcal{R} : \Phi_{u,r} > 0\}, \tag{2.45}$$

$$w^K_{k,r} = w^{K,\text{upper}}_{k,r}, \quad \forall\ k \in \mathcal{K}_r, r \in \{\mathcal{R} : \Phi_{u,r} > 0\}, \tag{2.46}$$

$$w^P_r >= 0.5 \sum_{g \in \mathcal{G}} \sum_{k \in \mathcal{K}_r} b^K_{g,k,r}, \quad r \in \{\mathcal{R} : \Phi_{u,r} > 0\}, \tag{2.47}$$

$$h^K_{1,r} \leq H^R_r - H^{K,\text{left}}_{u,r}, \quad \forall\ k \in \mathcal{K}_r, r \in \{\mathcal{R} : \Phi_{u,r} > 0\}, \tag{2.48}$$

$$h^K_{|\mathcal{K}_r|,r} \leq H^R_r - H^{K,\text{right}}_{u,r}, \quad \forall\ k \in \mathcal{K}_r, r \in \{\mathcal{R} : \Phi_{u,r} > 0\}, \tag{2.49}$$

$$w^K_{k,r} \geq W^{K,\text{left}}_{u,r}, \quad \forall\ k \in \mathcal{K}_r, r \in \{\mathcal{R} : \Phi_{u,r} > 0\}, \tag{2.50}$$

$$w^K_{|\mathcal{K}_r|,r} \geq W^{K,\text{right}}_{u,r}, \quad \forall\ k \in \mathcal{K}_r, r \in \{\mathcal{R} : \Phi_{u,r} > 0\}, \tag{2.51}$$

$$b^K_{g,k,r} \in \{0,1\}, \quad \forall\ g \in \mathcal{G}, k \in \mathcal{K}_r, r \in \{\mathcal{R} : \Phi_{u,r} > 0\}, \tag{2.52}$$

$$G^{\max}_{g,k,r} \geq x^G_{g,k,r} \geq 0, w^P_r \geq 0,$$

$$\text{and} \in \text{integer}, \forall\ g \in \mathcal{G}, k \in \mathcal{K}_r, r \in \{\mathcal{R} : \Phi_{u,r} > 0\}, \tag{2.53}$$

$$c^{K,\max}_{k,r}, h^K_{k,r}, w^F, w^K_{k,r}, w^{K,\text{upper}}_{k,r} \geq 0,$$

$$\forall\ k \in \mathcal{K}_r, r \in \{\mathcal{R} : \Phi_{u,r} > 0\}. \tag{2.54}$$

The objective function (2.37) seeks to minimize the total required width of the floor area, and the corresponding Constraint (2.38) determines the maximum used width of all rows. Constraint (2.39) ensures that every machine is allocated space on the floor. Next, Constraints (2.40) and (2.41) determine the maximum clearance requirement among all machines in every row for Constraint (2.42) to determine the occupied height of each used column. Therefore, Constraint (2.43) ensures the height of each column is less than the corresponding row height. Then, Constraint (2.44) determines the width occupied by each column in each row. Furthermore, Constraint (2.46) sets the found column widths as upper bounds for `S2`. Moreover, Constraint (2.47) determines the number of walkways required in row r. Additionally, Constraint (2.45) ensures at least one machine is allocated to row r if $\Phi_{u,r} > 0$. Constraints (2.48) – (2.51) set the size bounds for boundary columns if adjacent FAs of u have been allocated. Finally, Constraints (2.52) – (2.54) describe the decision variable definitions.

The purpose of `S2` is to minimize the space occupied by the boundary columns, k_r^B, in row $r \in \{\Phi_{u,r} > 0\}$ after `S1` has minimized the floor width. Note that k_r^B only represents the boundary columns between u and the un-allocated adjacent FAs. If both sides of the adjacent FAs in row r have not been allocated , k_r^B is the boundary column on the right side of u. That is, `S2` not only has all the constraints of `S1` except for Constraint (2.46), but also Constraint (2.56) to set upper bounds for the widths of each used column:

$$\text{(Model S2): Min} \quad \sum_{r \in \{\mathcal{R}: \Phi_{u,r} > 0, \Phi_{u,r} \pm 1 \in \Phi_{f'',r}, f'' \in (\mathcal{F} \setminus \mathcal{A})\}} w_{k_r^B,r}^{K,\text{upper}} h_{k_r^B,r}^{K} \tag{2.55}$$

$$\text{s.t.} \quad \text{Constraints (2.38) to (2.54)}$$

$$w_{k,r}^{K} = w_{k,r}^{K,\text{upper}}, \quad \forall\, k \in \mathcal{K}_r, r \in \{\mathcal{R} : \Phi_{u,r} > 0\}. \tag{2.56}$$

Note that, in practice, based on the product demands, it is possible for specific manufacturers to have a lower (upper) bound on how many machines from a FA appearing in multiple rows that should be allocated to each individual row. In such case, the bounds can be simply incorporated in the sub-problems as:

$$\text{The lower bound} \leq \sum_{\mathcal{G}} \sum \mathcal{K} x^{\mathcal{G}}_{g,k,r} \leq \text{The upper bound for particular row } r.$$

In S1 and S2, machines are allocated by order of descending clearance requirements to ensure minimum total side clearance consumption within columns. This property is proven below.

Property: The total clearance consumption for a certain set of machines in a column is minimized if the machines are arranged in the order of descending clearance requirement, and the minimum total side clearance requirements are equal to the summation of the clearance requirements on one side of all the machines in the column and the maximum the clearance on one side of these machines.

Proof. Without loss of generality, suppose there are N machines in a column, and all of the machines are numbered such that the clearance requirement $C^{M,S}_{m,o} > C^{M,S}_{n,o}$, $\forall\ m < n$, and $m, n \in N$. Then the total clearance requirement δ for the set is:

$$\begin{aligned}\delta &= C^{M,S}_{1,o} + \max\{C^{M,S}_{1,o}, C^{M,S}_{2,o}\} + \max\{C^{M,S}_{2,o}, C^{M,S}_{3,o}\} + ... + \max\{C^{M,S}_{N-1,o}, C^{M,S}_{N,o}\} + C^{M,S}_{N,o} \\ &= 2C^{M,S}_{1,o} + C^{M,S}_{2,o} + C^{M,S}_{3,o} + ... + C^{M,S}_{N-1,o} + C^{M,S}_{N,o}\end{aligned}$$

By contradiction, assume machines j and i are pairwise interchanged, $j < i$; then δ and the new total clearance requirement δ' are:

$$\begin{aligned}\delta =& C^{M,S}_{1,o} + \max\{C^{M,S}_{1,o}, C^{M,S}_{2,o}\} + ... + \max\{C^{M,S}_{j-1,o}, C^{M,S}_{j,o}\} + \max\{C^{M,S}_{j,o}, C^{M,S}_{j+1,o}\} + \max\{C^{M,S}_{j+1,o}, C^{M,S}_{j+2,o}\} \\ &+ ... + \max\{C^{M,S}_{i-1,o}, C^{M,S}_{i,o}\} + \max\{C^{M,S}_{i,o}, C^{M,S}_{i+1,o}\} + ... + \max\{C^{M,S}_{N-1,o}, C^{M,S}_{N,o}\} + C^{M,S}_{N,o} \\ =& 2C^{M,S}_{1,o} + C^{M,S}_{2,o} + ... + C^{M,S}_{j-1,o} + C^{M,S}_{j,o} + C^{M,S}_{j+1,o} + C^{M,S}_{j+2,o} + ... + C^{M,S}_{i-2,o} + C^{M,S}_{i-1,o} + C^{M,S}_{i,o} + C^{M,S}_{i+1,o} \\ &+ ... + C^{M,S}_{N-1,o} + C^{M,S}_{N,o}\end{aligned}$$

$$
\begin{aligned}
\delta' =& C^{M,S}_{1,o} + \max\{C^{M,S}_{1,o}, C^{M,S}_{2,o}\} + ... + \max\{C^{M,S}_{j-1,o}, C^{M,S}_{i,o}\} + \max\{C^{M,S}_{i,o}, C^{M,S}_{j+1,o}\} + \max\{C^{M,S}_{j+1,o}, C^{M,S}_{j+2,o}\} \\
&+ ... + \max\{C^{M,S}_{i-1,o}, C^{M,S}_{j,o}\} + \max\{C^{M,S}_{j,o}, C^{M,S}_{i+1,o}\} + ... + \max\{C^{M,S}_{N-1,o}, C^{M,S}_{N,o}\} + C^{M,S}_{N,o} \\
=& 2C^{M,S}_{1,o} + C^{M,S}_{2,o} + ... + C^{M,S}_{j-1,o} + 2C^{M,S}_{j+1,o} + C^{M,S}_{j+2,o} + ... + C^{M,S}_{i-2,o} + 2C^{M,S}_{j,o} + C^{M,S}_{i+1,o} + ... + C^{M,S}_{N-1,o} \\
&+ C^{M,S}_{N,o}
\end{aligned}
$$

Moreover, the difference between δ' and δ is:

$$
\begin{aligned}
\delta' - \delta &= (C^{M,S}_{j+1,o} + 2C^{M,S}_{j,o}) - (C^{M,S}_{j,o} + C^{M,S}_{i-1,o} + C^{M,S}_{i,o}) \\
&= (C^{M,S}_{j+1,o} + C^{M,S}_{j,o}) - (C^{M,S}_{i-1,o} + C^{M,S}_{i,o})
\end{aligned}
$$

It is clear that $C^{M,S}_{j+1,o} > C^{M,S}_{k-1,o}$ and $C^{M,S}_{j,o} > C^{M,S}_{k,o}$, so $\delta' - \delta > 0$. Therefore $\delta' > \delta$. This contradicts the total clearance requirement and is minimized by sorting in an ascending/descending order of the clearance requirements in a column, thereby completing the proof of the property. Additionally, in this sorting procedure, the minimum total side clearance requirement of δ is equal to the summation of the one-side clearance requirements of all machines in the column and the maximum one-side clearance of these machines.

2.4.4 Functional area interlocking

At this stage of the procedure, FA interlocking is performed by combining the boundary columns of u and the allocated adjacent FAs. We begin by defining the notation required in column integration for functional area interlocking. Let Z^{ϕ}_{r} represent the set of allocated machine columns in row r, and let $z^{\phi}_{k,r,l} \subseteq Z^{\phi}_{r}$ represent the lth machine (l= 1,...,$|z^{\phi}_{k,r}|$) that has been assigned to column k of row r in the constructed layout, $\forall\ k \in \mathcal{K}_r, r \in \{\mathcal{R} : \Phi_{f,r} > 0\}$, and $f \in \mathcal{A}$. $z^{\phi}_{k,r,l}$ cumulatively takes over the solutions of $x^{G}_{g,k,r}$ from each iteration until all FAs are allocated. Moreover, K'_r (K''_r) represents the number of assigned columns in row $r \in \{\mathcal{R} : \Phi_{u,r} > \Phi_{f',r} > 0\}$ ($r \in \{\mathcal{R} : \Phi_{f',r} > \Phi_{u,r} > 0\}$) for all $f' \in \mathcal{A} \backslash u$, and K'''_r represents the number of columns for u in $r \in \{\mathcal{R} : \Phi_{u,r} > 0\}$.

For functional area interlocking, if the adjacent FAs of u have been allocated ($\Phi_{u,r} - 1 \in \Phi_{f',r}$ or $\Phi_{u,r} + 1 \in \Phi_{f',r}$), boundary columns of u and the adjacent FAs are pieced together (lines 3–26 in Algorithm 2). Additionally, if u is the last FA in row r ($\Phi_{u,j}$ =max$\{\Phi_{f,j},\ \forall\ f \in \mathcal{F}\}$) and the adjacent FA has not been allocated ($\Phi_{u,r} - 1 \notin \Phi_{f',r}$), the last column of u in row r is relocated to the place right before the first column of u in row r for subsequent functional area interlocking. As the last column usually takes up the least space according to `S2`, the remaining space within the column could be used for FA interlocking by relocation. Then, based on the proven Property, the machines in each column are sorted in descending order in $C_g^{G,S}$, but breaks tie by selecting machines with longer W_g^G. This means that the same side clearance machines with longer widths are allocated at the front in a column, and later on in the beginning of Algorithm 3, even machine columns will be sorted in reverse order, which makes it possible to match two adjacent reverse up-side-down ladder-shaped columns for optimal space saving by machine interlocking.

2.4.5 Physical arrangement for machine interlocking

Algorithm 3 is implemented to ascertain the physical locations of machines with the considerations of machine interlocking. First, machines within each $z_{k,q}^{\phi}$ are sorted in reverse order for all even numbers of k. As mentioned, this sorting procedure makes it possible to match two adjacent reverse up-side-down ladder-shaped columns for optimal space saving by machine interlocking. Second, based on Z_r^{ϕ}, the x and y coordinates on the bottom corner of the l^{th} machine in column k of row r, $x_{k,r,l}^{\phi}$ and $y_{k,r,l}^{\phi}$, are determined. Next, machines in even columns are moved in the upward direction first, then all machines are moved toward the left as much as possible without violating the physical constraints of the MRMIC problems. This improves space utilization by machine interlocking as well. Then the shape of the boundary columns within u is set as bounds for boundary columns of un-allocated adjacent FAs. In the last part of Algorithm 3, the space occupied by u, denoted by $A_{u,r}^{O}$, is determined by

Algorithm 2 Functional area interlocking

function FAI(u,$x^G_{g,k,r}$,Z^ϕ_r)
  Find k'_r, k''_r, and k'''_r
  $Z^\phi_r \leftarrow \{z^\phi_{1,r}, ..., z^\phi_{k'_r,r}, z^\phi_{k'_r+1,r} = \varnothing, ..., z^\phi_{k'_r+k'''_r,r} = \varnothing, z^\phi_{k'_r+k'''_r+1,r}, ..., z^\phi_{k'_r+k''_r+k'''_r,r}\}$,
    $\forall r \in \{\mathcal{R} : k'''_r > 0\}$
  for $j \in \{\mathcal{R} : \Phi_{u,r} > 0\}$ **do**
    for $i = 1$ to k'''_j **do**
      if $x^G_{g,i,j} > 0$ **then**
        for $l = 1$ to $x^G_{g,i,j}$ **do**
          if $k'_j > 0$ **then**
            $z^\phi_{k'_j+i-1,j,|z^\phi_{k,r}|+1} \leftarrow g$
          else
            $z^\phi_{i,j,|z^\phi_{k,r}|+1} \leftarrow g$
          end if
        end for
      end if
    end for
    if $(\Phi_{u,r} - 1 \in \Phi_{f',r})$ **then**
      $z^\phi_{k'_j,j} \leftarrow z^\phi_{k'_j,j} \cup z^\phi_{k'_j+1,j}$; $Z^\phi_j \leftarrow Z^\phi_j \backslash z^\phi_{k'_j+1,j}$
    end if
    if $(\Phi_{u,r} + 1 \in \Phi_{f',r})$ **then**
      $z^\phi_{k'_j+k'''_j,j} \leftarrow z^\phi_{k'_j+k'''_j,j} \cup z^\phi_{k'_j+k'''_j+1,j}$; $Z^\phi_j \leftarrow Z^\phi_j \backslash z^\phi_{k'_j+k'''_j+1,j}$
    end if
    if $(\Phi_{u,j} = \max\{\Phi_{f,j},\ \forall\ f \in \mathcal{F}\})$ & $(\Phi_{u,j} - 1 \notin \Phi_{f',j})$ **then**
      Move $z^\phi_{k'_j+k'''_j,j}$ into the position right after $z^\phi_{k'_j,j}$ in Z^ϕ_j
    end if
  end for
  Sort machine order within each $z^\phi_{k,r}$ by descending order of $C^{G,S}_g$
    (breaking tie by W^G_g)
  return Z^ϕ_r
end function

the summation of all non-boundary occupied column space, walkway space, machine and clearance requirement space in the boundary columns of u.

Although space saving by machine interlocking depends on the shapes of every two adjacent columns, limited improvements were observed in our preliminary tests of local search methods for different column pairing. This indicates that the proposed heuristic does not include a mechanism for finding better combinations of column pairing within the same rows.

2.4.6 Neighborhood search with relaxing pre-determined relative locations of FAs

In lines 1–14 in Algorithm 1, we introduced the procedure to generate a layout plan when the relative location of FAs is pre-determined. This is because in practice, semiconductor manufacturing plants have pre-determined relative locations for their equipment in correspondence with their production process. However, the proposed heuristic could be implemented in the case of flexible relative locations. In this case, an initial relative location of FAs is required to determine the facility layout plan by executing lines 1–14 in Algorithm 1. Subsequently, an iterative neighborhood search procedure is proposed in Algorithm 4 to improve the layout plan by relaxing the given initial relative location.

The neighborhood search procedure only modifies the layouts of the longest row and the shortest row in the layout plan iteratively, starting with an initially empty facility for these two rows. The idea is to balance the row widths by moving the column with the most white space from the longest row to the shortest row. Let $\mathcal{K}'$ represent the column set from a FA in the longest row, and let $\mathcal{K}'$ be sorted from the column with the most white space to that with the least white space. The white space represents the space that is wasted when calculating the total space consumption of a column (if there is no adjacent column) or of two adjacent columns minus the continuous space approximation of the machines within the column(s). The white space of each one of the two adjacent interlocking column is

Algorithm 3 Physical Arrangement for machine interlocking

function $\texttt{PA}(H^{K,\text{left}}_{f,r}, H^{K,\text{right}}_{f,r}, u, W^{K,\text{left}}_{f,r}, W^{K,\text{left}}_{f,r}, Z^{\phi}_{r})$
  Sort $z^{\phi}_{k,q}$ backwards for all $k \in$ the columns on the right edge of a walkway
  for $j \in \mathcal{R}$ **do**
    for $i \in \mathcal{K}_j$ **do**
      if $z^{\phi}_{i,j} \neq \varnothing$ **then**
        for $l = 1$ to $|z^{\phi}_{i,j}|$ **do**
          if $l \neq 1$ **then**
            $y^{\phi}_{i,j,l} \leftarrow y^{\phi}_{i,j,(l-1)} + W^{G}_{z^{\phi}_{i,j,(l-1)}} + \max\{C^{G,S}_{z^{\phi}_{i,j,l}}, C^{G,S}_{z^{\phi}_{i,j,(l-1)}}\}$
          else
            $y^{\phi}_{i,j,l} \leftarrow C^{G,S}_{z^{\phi}_{i,j,l}}$
          end if
          if $i \in$ even numbers **then**
            $x^{\phi}_{i,j,l} \leftarrow \max\big\{x^{\phi}_{i-1,j,n} + W^{G}_{z^{\phi}_{i-1,j,n}},\ \forall\, n \in \{1, ..., |z^{\phi}_{i-1,j}|\}\big\} + W^{P}$
          else
            if $i = 1$ **then**
              $x^{\phi}_{i,j,l} \leftarrow \max\big\{C^{G,B}_{z^{\phi}_{i,j,n}} + W^{G}_{z^{\phi}_{i,j,n}},\ \forall\, n \in \{1, ..., |z^{\phi}_{i,j}|\}\big\} - W^{G}_{z^{\phi}_{i,j,l}}$
            else
              $x^{\phi}_{i,j,l} \leftarrow \max\big\{x^{\phi}_{i-1,j,n} + W^{G}_{z^{\phi}_{i-1,j,n}} + C^{G,B}_{z^{\phi}_{i-1,j,n}},\ \forall\, n \in \{1, ..., |z^{\phi}_{i-1,j}|\}\big\}$
              $+\max\big\{W^{G}_{z^{\phi}_{i,j,n}},\ \forall\, n \in \{1, ..., |z^{\phi}_{i,j}|\}\big\} - W^{G}_{z^{\phi}_{i,j,l}}$
            end if
          end if
        end for
      end if
      Pull machines in even columns toward upward as much as possible
      Pull all machines toward left as much as possible
    end for
    if $(\Phi_{u,j} = \max\{\Phi_{f,j},\ \forall\, f \in \mathcal{F}\})$ & $(\Phi_{u,j} - 1 \notin \Phi_{f',j})$ **then**
      $i' \leftarrow$ the first column of I in row j
      $H^{K,\text{right}}_{\Phi_{u,j}-1,j} \leftarrow (y^{\phi}_{i',j,|z^{\phi}_{i',j}|} + C^{G,S}_{z^{\phi}_{i',j,|z^{\phi}_{i',j}|}})$
      $W^{K,\text{right}}_{\Phi_{u,j}-1,j} \leftarrow \max\big\{C^{G,B}_{z^{\phi}_{i',j,n}} + W^{G}_{z^{\phi}_{i',j,n}},\ \forall\, n \in \{1, ..., |z^{\phi}_{i',j}|\}\big\}$
    else
      $i' \leftarrow$ the last column of u in row j
      $H^{K,\text{left}}_{\Phi_{u,j}+1,j} \leftarrow (y^{\phi}_{i',j,|z^{\phi}_{i',j}|} + C^{G,S}_{z^{\phi}_{i',j,|z^{\phi}_{i',j}|}})$
      $W^{K,\text{left}}_{\Phi_{u,j}+1,j} \leftarrow \max\{C^{G,B}_{z^{\phi}_{i',j,n}} + W^{G}_{z^{\phi}_{i',j,n}},\ \forall\, n \in \{1, ..., |z^{\phi}_{i',j}|\}\}$
    end if
    Find $A^{O}_{f',r}$ and $A^{O}_{u,r}$; $A^{O}_{f',r} \leftarrow A^{O}_{f',r} \cup A^{O}_{u,r}$
  end for
  return $A^{O}_{f',r}$,
end function

Algorithm 4 Neighborhood search with relaxing pre-determined relative locations of FAs

function `NS`($A^{O}_{f',r}$, $H^{K,\text{left}}_{f,r}$,$H^{K,\text{right}}_{f,r}$,$\mathcal{U}$,$W^{K,\text{left}}_{f,r}$,$W^{K,\text{left}}_{f,r}$,Z^{ϕ}_{r})
    repeat
        Initialize the machine column list $\mathcal{K}'$ in the longest row r of the whole layout
        Initialize $\mathcal{U}'$, $\mathcal{A} = \{$FA which the column is from$\}$, and i=1
        while i$\leq |\mathcal{K}'|$ or no any better solution is found **do**
            Move the i^{th} machine column in $\mathcal{K}'$ into the shortest row
            $u \leftarrow \mathcal{A}$
            $r \leftarrow$ The longest row; Then execute lines 5-12 in Algorithm 1
            if the i^{th} column in $\mathcal{K}'$ from the FA appearing in the shortest row **then**
                $r \leftarrow$ The shortest row then execute lines 5-12 in Algorithm 1
                for all $u \in \mathcal{U}'$ **do**
                    Execute lines 4-13 in Algorithm 1
                end for
            else
                for j=1 to # of FAs in the shortest row +1 **do**
                    Update $\mathcal{U}'$ by setting the i^{th} column as the j^{th} FA in the shortest row
                    $r \leftarrow$ The shortest row; Then execute lines 5-12 in Algorithm 1
                    for all $u \in \mathcal{U}'$ **do**
                        Execute lines 4-13 in Algorithm 1
                    end for
                end for
            end if
            i$\leftarrow$i+1
        end while
    until (Stop)
    return Best found $A^{O}_{f',r}$,Z^{ϕ}_{r}
end function

measured by proportionally dividing the total white space of these two columns based on the continuous space approximation of these two. The same rule applies when determining the white space of a column with machines from two FAs (the boundary column between two FAs).

An example in which $\mathcal{K}'$ is determined is provided below. Suppose there are only three columns in the longest row, ${\mathcal{K}'}_1$ with white space of five ft^2, ${\mathcal{K}'}_2$, and ${\mathcal{K}'}_3$, where ${\mathcal{K}'}_2$, and ${\mathcal{K}'}_3$ share white space of 20 ft^2. ${\mathcal{K}'}_2$ only contains machines from the Films FA with a continuous space approximation of 100 ft^2. ${\mathcal{K}'}_3$ contains machines from the Films FA with a continuous space approximation of 60 ft^2 and from the Diffusion FA with a continuous space approximation of 40 ft^2. Then $\mathcal{K}' = \{{\mathcal{K}'}_2, {\mathcal{K}'}_3 \text{ from Films }, {\mathcal{K}'}_1, {\mathcal{K}'}_3 \text{ from Diffusion }\}$.

As described, once $\mathcal{K}'$ is determined, we verify whether there is any improvement by moving the column from $\mathcal{K}'$ to the shortest row one at a time until either a better solution is found or the whole $\mathcal{K}'$ is tested. If the column is from an FA that appears in the shortest row, the column is assigned into the FA and we initialize the relative location of FA, $\mathcal{U}'$, which includes the FAs and the machines from only the longest and the shortest rows. We re-determine the layout of these two rows to establish whether there is any improvement by relocating the column to the shortest row. If the column is not from the FA that appears in the shortest row, we re-determine the layout of these two rows with different $\mathcal{U}'$ generated by inserting the column into different positions of the relative location in the shortest row. For example, assuming the column is from the Films FA, and the relative location of the FA in the shortest row is $\{\text{Etch}, \text{Diffusion}\}$, we re-determine the layout with the different relative location of FAs in the shortest row as $\{\text{Films}, \text{Etch}, \text{Diffusion}\}$, $\{\text{Etch}, \text{Films}, \text{Diffusion}\}$, and $\{\text{Etch}, \text{Diffusion}, \text{Films}\}$ to establish which relative location is associated with the best re-layout plan. Then the neighborhood search procedure iteratively executes lines 4–13 in Algorithm 1 to determine whether there are better solutions.

2.5 Numerical analysis

A numerical analysis was conducted to evaluate the performance of our heuristic. To the best of our knowledge, past approaches to solve bin-packing problems cannot be used to solve the same problem because they assumed the sizes of bins were given. This differs from our problem in which the sizes of columns are decision variables. Furthermore, there is a lack of applicable methodologies for the FLP. Thus, a lower bound is provided in Section 2.5.1 to evaluate the performance of the proposed heuristic.

Next, problems of a practical size were generated as test beds for our approach in Section 2.5.2. As explained, the proposed heuristic can solve the MRMIC for either fixed relative locations or for flexible relative locations of FAs by incorporating the neighborhood search procedure. Hence, the computational experiment in this section consists of two phases. The first phase illustrates the performance of the proposed heuristic in solving the test problems when the relative location of FAs is pre-determined. Then, the second phase, the neighborhood search procedure, relaxes the relative location of FAs to determine whether a more optimal solution could be found based on the solution provided in the first phase.

All experiments were conducted on a PC with a core Intel i5-2410m processor running Microsoft Windows 7 in 64-bit mode. The heuristic was programmed in MATLAB, and the sub-problems were solved by using the Gurobi Optimizer version 5.6.3.

2.5.1 The proposed lower bound

Considering a continuous lower bound for MRMIC,

$$\mathcal{B} = \sum_{m\in\mathcal{M}_f,\ f\in\mathcal{F}} (H_m^M + C_m^{M,S})[W_m^M + 0.5(C_m^{M,B} + W^P)]$$

is the sum of A_f, $\forall\ f \in \mathcal{F}$. Let w' =min$\{W_m^M, \forall\ m \in \mathcal{M}_f,\ f \in \mathcal{F}\}$ be the shortest width of all machines. Then a simple version of the MRMIC layout problem, `BP1`, which could be considered as a variant of the 1-D bin packing problem, is to cut the width of every machine down to w', while retaining the original heights and clearances of the MRMIC. Thus, the cut-off continuous space is represented as:

$$\mathcal{B}^{\text{cutoff}} = \mathcal{B} - \sum_{\forall m \in \mathcal{M}_f,\ f \in \mathcal{F}} (H_m^M + C_m^{M,S})[w' + 0.5(C_m^{M,B} + W^P)] = \sum_{\forall m \in \mathcal{M}_f,\ f \in \mathcal{F}} (H_m^M + C_m^{M,S})(W_m^M - w')$$

Suppose the optimal space of the MRMIC is $O^{\texttt{MRMIC}}$, and the optimal space of the corresponding BP1 is $O^{\texttt{BP1}}$. Thus, a better lower bound is proposed as $O^{\texttt{BP1}} + \mathcal{B}^{\text{cutoff}}$.

Theorem: $O^{\texttt{MRMIC}} \geq O^{\texttt{BP1}} + \mathcal{B}^{\text{cutoff}}$

Proof. Without loss of generality, suppose the optimal occupied space of a column k in row r in MRMIC is

$$O_{k,r}^{\texttt{MRMIC}} \geq \mathcal{B}_k + \mathcal{S}_{m_1,k} + \mathcal{S}_k^{\text{waste}},$$

where

$$\mathcal{B}_k = \sum_{\forall m \in M_k} (H_m^M + C_m^{M,S})[W_m^M + 0.5(C_m^{M,B} + W^P)]$$

represents the continuous required space for all $m \in M_k$. $M_k \in \mathcal{M}_f$ indicates all machines in column k, where $f \in \mathcal{F}$. $\mathcal{S}_{m_1,k} = C_{m_1}^{M,S}[W_{m_1}^M + 0.5(C_{m_1}^{M,B} + W^P)]$ represents one side clearance and its connected walkway space of machine $m_1 \in M_k$ (similar to the proven property, the total amount of required space in a column is at least the sum of $\mathcal{B}_k$ and a side clearance of a machine, which is determined by the MRMIC model). $\mathcal{S}_k^{\text{waste}}$ is the minimum white space besides the space occupied by machines and clearances within column k. Note that

$$\mathcal{S}_k^{\text{waste}} \geq (H_r^R - h_{k,r}^K) \min\{W_m^M,\ \forall\ m \in M_k\}.$$

Then the optimal occupied space of column k in row r for problem BP1 is

$$O_{k,r}^{\texttt{BP1}} = \mathcal{B}_k^{\texttt{BP1}} + \mathcal{S}'_{m'_1,k} + \mathcal{S}_k'^{\text{waste}},$$

where

$$\mathcal{B}_k^{\texttt{BP1}} = \sum_{\forall m' \in M'_k} (H_{m'}^M + C_{m'}^{M,S})[w'_k + 0.5(C_{m'}^{M,B} + W^P)]$$

represents the continuous lower bound of the required space for all $m' \in M'_k$. M'_k represents all the corresponding machines from M_k, but only cuts off their width down to w'_k =min$\{W_m^M, \forall\ m \in M_k\}$. In such a case,$H_{m'}^M$, $C_{m'}^{M,S}$, and $C_{m'}^{M,B}$ are still equal to H_m^M, $C_m^{M,S}$, and $C_m^{M,B}$ for all $m' \in M'_k$, and $m \in M_k$, respectively. Moreover,

$$\mathcal{S}'_{m'_1,k} = C_{m_1}^{M,S}[w' + 0.5(C_{m_1}^{M,B} + W^P)] \leq \mathcal{S}_{m_1,k},\ m_1 \in M_k, m'_1 \in M'_k,$$

and

$$\mathcal{S}_k'^{\text{waste}} = (H_r^R - h_{k,r}^K)w' \leq \mathcal{S}_k^{\text{waste}}.$$

Additionally, the cut-off continuous space is represented as

$$\mathcal{B}_k^{\text{cutoff}} = \mathcal{B}_k - \mathcal{B}_k^{\texttt{BP1}} = \sum_{\forall m \in M_k} (H_m^M + C_m^{M,S})(W_m^M - w'_k) \geq 0.$$

Therefore,

$$\begin{aligned} O_{k,r}^{\texttt{MRMIC}} - O_{k,r}^{\texttt{BP1}} &= \mathcal{B}_k + \mathcal{S}_{m_1,k} + \mathcal{S}_k^{\text{waste}} - (\mathcal{B}_k^{\texttt{BP1}} + \mathcal{S}'_{m'_1,k} + \mathcal{S}_k'^{\text{waste}}) \\ &= (\mathcal{B}_k - \mathcal{B}_k^{\texttt{BP1}}) + (\mathcal{S}_{m_1,k} - \mathcal{S}'_{m'_1,k}) + (\mathcal{S}_k^{\text{waste}} - \mathcal{S}_k'^{\text{waste}}) \geq 0, \end{aligned}$$

so $O_{k,r}^{\texttt{MRMIC}} - O_{k,r}^{\texttt{BP1}} - (\mathcal{B}_k - \mathcal{B}_k^{\texttt{BP1}}) = (\mathcal{S}_{m_1,k} - \mathcal{S}'_{m'_1,k}) + (\mathcal{S}_k^{\text{waste}} - \mathcal{S}_k'^{\text{waste}}) \geq 0.$

We find $O^{\mathtt{MRMIC}}_{k,r} \geq O^{\mathtt{BP1}}_{k,r} + \mathcal{B}^{\text{cutoff}}_{k}$ for any column k in any row r. Hence, $O^{\mathtt{MRMIC}} \geq O^{\mathtt{BP1}} + \mathcal{B}^{\text{cutoff}}$. This lower bound could be found in polynomial time, and is approximately 2% better than simple LP relaxation of the MRMIC based on our preliminary results by considering $\mathcal{S}'_{m'_1,k}$ and $\mathcal{S}'^{\text{waste}}_{k}$ for machine m'_1 in column k.

2.5.2 Computational results

In the absence of applicable benchmark problems in the literature, test problems were created to evaluate the performance of the proposed approach for a variety of layout design parameters. The manufacturing process with associated FAs may be briefly introduced as follows. Initially, the films process (FIL) involves coating materials onto the wafer, and then the lithography process (LIT) patterns the un-coated parts of the wafer. The chemical-mechanical planarization process (CMP), wet-etch process (WET), and etch process (ETC) remove coated materials to obtain the desired electrical properties by the diffusion process (DIF) and the implant process (IMP) with dopants. Finally, the metrology process (METRO) is frequently used between the various processing steps to verify any damage that has occurred. In Table 2.6, we generated the machine data for each FA based on uniform distributions, although the range of each distribution is assumed based on practical data. Additionally, we specified 30 inches, which is adjusted based on the information from our industrial partner, for back clearances of all machines.

Table 2.7 lists four parameters, including the physical structure of the floor, the size of walkways, and the relative locations of FAs. Two of the relative locations of FAs were adjusted by using practical information and data from previous studies, respectively. The first relative location is DIF $\rightarrow$ IMP $\rightarrow$ FIL $\rightarrow$ ETC for odd rows but CMP $\rightarrow$ WET $\rightarrow$ FIL $\rightarrow$ LIT for even rows. The second type is FIL $\rightarrow$ LIT $\rightarrow$ ETC $\rightarrow$ METRO for odd rows from Susto et al. [24] but FIL $\rightarrow$ LIT $\rightarrow$ ETC $\rightarrow$ IMP$\rightarrow$ CMP for even rows, from Khan et al. [15]. The proposed approach was tested for a variety of layout structures by conducting experiments by varying the number of rows, which were mixed with different row

Table 2.6: Problem generation

Functional Area	# of Machines in Each Group	# of Machine Groups	Machine Width (inches)	Machine Height (inches)	Side Clearance (inches)
CMP	U(100,120)	U(1,3)	U(20,100)	U(70,120)	30
DIF	U(20,30)	U(8,12)	U(5,60)	U(40,150)	3,30
ETC	U(15,25)	U(8,10)	U(18,22)	U(60,150)	30
FIL	U(40,50)	U(8,12)	U(10,100)	U(60,150)	30
IMP	U(50,80)	U(2,4)	U(5,50)	U(100,150)	30
LIT	U(80,100)	U(2,4)	U(50,100)	U(80,100)	30
METRO	U(25,75)	U(1,3)	U(50,100)	U(80,100)	30
WET	U(20,30)	U(12,14)	U(2,50)	U(80,100)	30

heights. For example, in the experiments we enumerated four combinations of a two-row layout mixed with row heights, including layouts with two long-height rows, two short-height rows, and a combination of both short- and long-height rows. Thus, there were four types of two-row layouts, six types of three-row layouts, and nine types of four-row layouts for all the combinations of row heights and row numbers. Each combination was tested with two types of walkway widths and two relative locations; hence, 76 test problems were provided in total for the experiments. Due to the required computational time of solving `S1` varies from the context of each problem, a cut-off time could be set as t_u^{cutoff} to run `S1` by the amount of machines in all $f \in \mathcal{U}$, where $u \in \mathcal{U}$. Because of the found $w_{k,r}^{K,\text{upper}}$ by `S1`, computational time of `S2` is ignorable. Let T^{limit} be the computational time limit for solving the whole MRMIC problem, and $N_{g,f}^F$ represents N_g^G in functional area f, for all $f \in \mathcal{U}$. Then $t_u^{\text{cutoff}} = T^{\text{limit}} \frac{\prod_{g \in \mathcal{G}} N_{g,u}^F}{\sum_{f \in \mathcal{U}} \prod_{g \in \mathcal{G}} N_{g,f}^F}$.

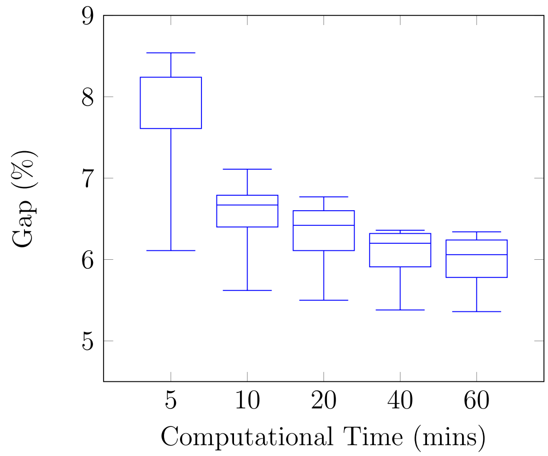
The computational result of the first phase is shown in Figure 2.5, which provides box plots that summarize the spread of the gaps, as measured by the percentage difference between our solution and the proposed lower bound. As shown in the figure, the results indicate that the proposed approach provides layout solutions with single percentage gaps in a reasonable runtime for all structure parameters. Not surprisingly, the layout solutions with fewer

Table 2.7: Parameters of layout configuration

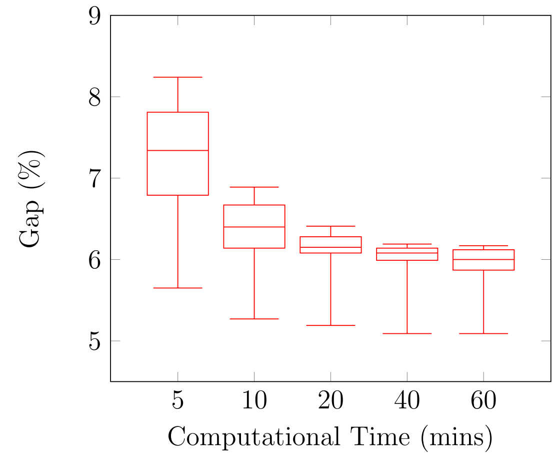
Parameter	Values	
# of rows	2, 3, 4	
Row height (inches)	Short:U(800,1200), Long:U(2800,3200)	
Width of Walkway (inches)	80, 90	
Relative Location of FA	1:	2:
	Odd Rows:D→I→F→E	Odd Rows:F→L→E→M [24]
	Even Rows:C→W→F→L	Even Rows:F→L→E→I→C [15]

rows yielded better results than the layouts with more rows within a short runtime, because the complexity of the proposed IP models grows when solving problems with additional rows. Thus, the gaps that were found for solving the problem with more rows take a longer time to converge because of the increasing number of decision variables in the IP models. As shown in the figure, the gaps converge within a runtime of 60 minutes on average.

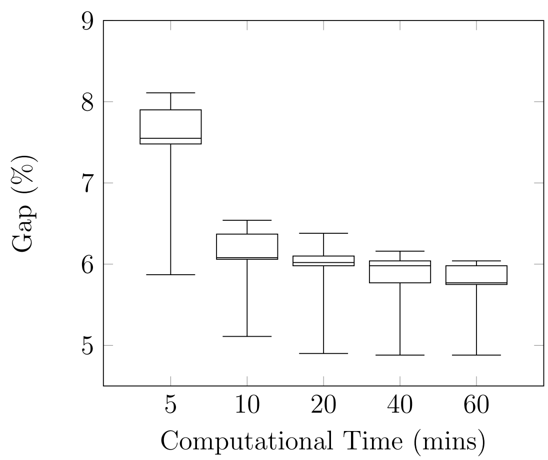
Figure 2.6 shows the result of the second phase experiment, which represents the percentage gaps between the proposed lower bound and our solutions obtained by incorporating the neighborhood search procedure. In this phase, we applied the procedure to determine whether we could improve the solution obtained after a 60-minute runtime in the first phase by relaxing the relative location of FAs. We also set the additional runtime for the second phase as 60 minutes. This is because solving the IP models with the upper bound obtained in the first phase during the neighborhood search procedure actually reduces the runtime in the second phase . Spending more time to determine a better upper bound in the first phase accelerates solution of the IP models in the second phase. Hence, we extended the runtime from 60 minutes to 120 minutes for the neighborhood search procedure instead of applying the neighborhood search procedure to improve the solution of the first phase obtained after 10 minutes runtime in the first phase. As shown in the figure, the additional 10 minutes required by relaxing the relative locations of FAs only led to a slight improvement in some of the layouts with three- and four-row structures. The common property of these layouts is that they have a last column in the longest row that does not use the space effectively before



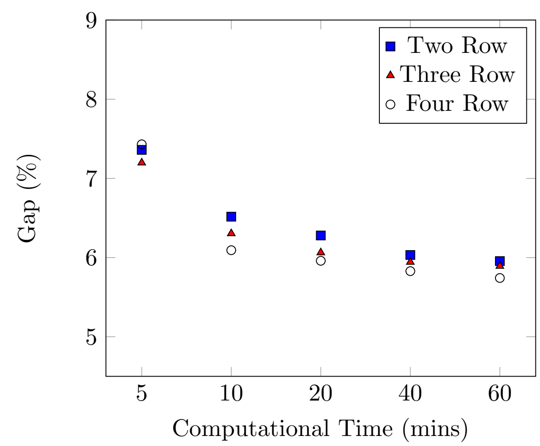
(a) Two row layout



(b) Three row layout



(c) Four row layout

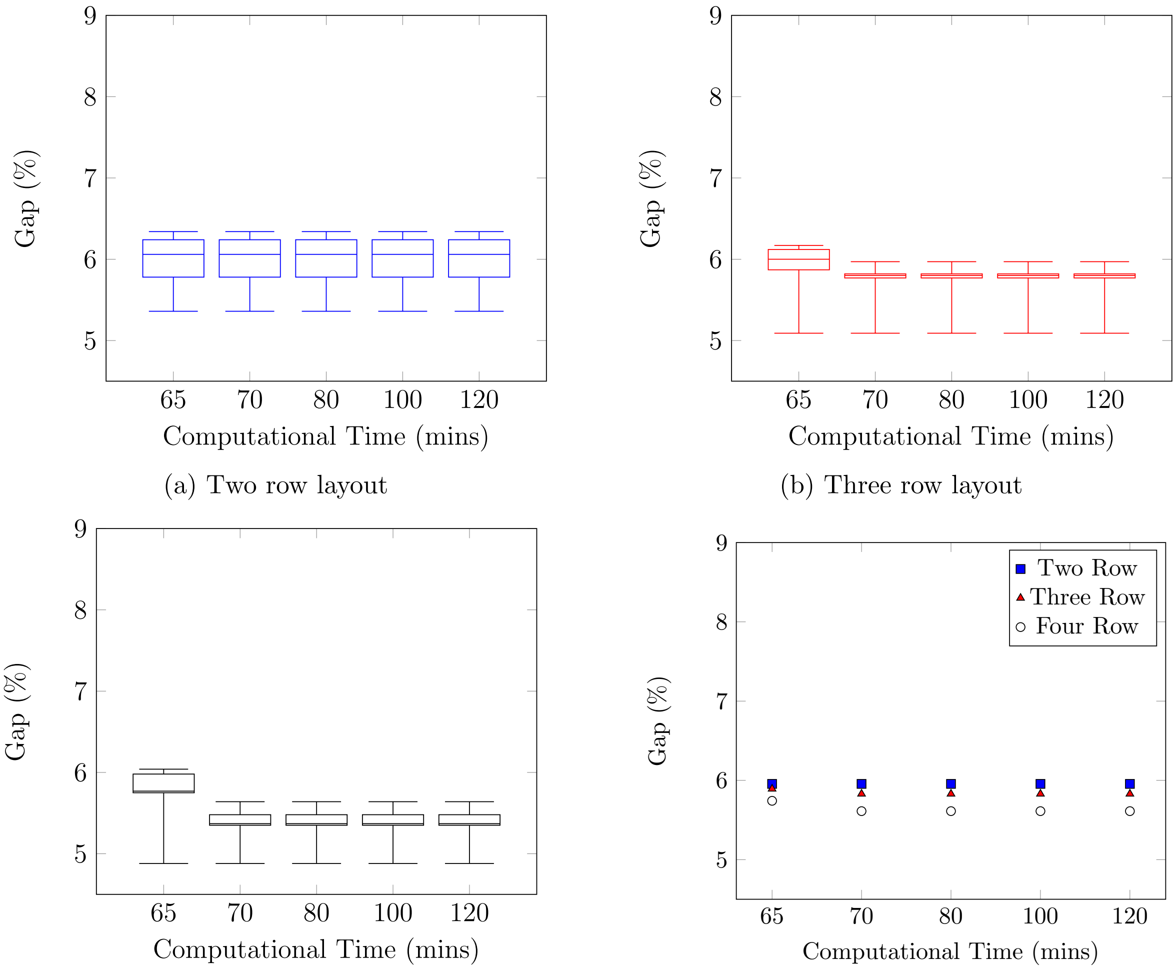


(d) Average gaps

Figure 2.5: Convergence of computational results in the first phase

relaxing the relative location, and the neighborhood search procedure is able to obtain a better layout by moving the column with the most white space from the longest row to the shortest row if the relative location is relaxed. The results confirm the performance of the proposed heuristic in the first phase under the relative location constraint, and demonstrate the second phase neighborhood search is able to improve the solution in a reasonable time.

As a result, both better lower bound and near-optimal layout solutions could be found in a limited time, indicating that the feasibility of fitting all required machines into a constrained facility could be determined by the proposed approach. The proposed approach helps to provide realistic information for future production planning while building or re-modeling a



(a) Two row layout
(b) Three row layout
(c) Four row layout
(d) Average gaps

Figure 2.6: Convergence of computational results in the second phase

fab. Furthermore, given machine sets generated by possible future demands, the feasibility analysis supports business decision-making in terms of long-term supply commitments to the customers.

2.6 Conclusions and future research

This paper defines an MRMIC FLP for semiconductor manufacturing, and presents a heuristic approach together with corresponding mathematical programming models for the problem. Unlike previous research, both machines and FAs were considered for the layout arrangement. In this research, the MRMIC layout structure was implemented. Moreover, space utilization, which is more important for semiconductor manufacturing, is considered as the objective. As a result, measured by the proposed lower bound, high-quality solutions could be found by the proposed approach in a limited time.

Although we focused on a feasibility analysis to locate all required machines within a constrained facility given the relative locations of FAs, interesting practical applications of the proposed solution exist for future study. This research considers space utilization as the objective for the facility layout plan on both FA and machine levels, but other objective functions, such as minimizing material handling flows, could be considered in future research. Another interesting future research objective would be to analyze the feasibility when the required machine sets are changed as a function of time, thus considering the trade-off between space utilization and re-layout costs for the proposed solution.

Bibliography

[1] R. Alvarez-Valdes, F. Parreño, and J.M. Tamarit. A GRASP/Path relinking algorithm for two- and three-dimensional multiple bin-size bin packing problems. *Computers & Operations Research*, 40(12):3081–3090, 2013.

[2] R.A. Arapoglu, B.A. Norman, and A.E. Smith. Locating input and output points in facilities design - A comparison of constructive, evolutionary, and exact methods. *IEEE Transactions on Evolutionary Computation*, 5(3):192–203, 2001.

[3] I. Castillo and T. Westerlund. An ϵ-accurate model for optimal unequal-area block layout design. *Computers & Operations Research*, 32(3):429–447, 2005.

[4] I. Castillo, J. Westerlund, S. Emet, and T. Westerlund. Optimization of block layout design problems with unequal areas: A comparison of MILP and MINLP optimization methods. *Computers & Chemical Engineering*, 30(1):54–69, 2005.

[5] J. Chae and B.A. Peters. Layout design of multi-bay facilities with limited bay flexibility. *Journal of Manufacturing Systems*, 25(1):1–11, 2006.

[6] J.C. Chen, R. Dai, and C. Chen. A practical fab design procedure for wafer fabrication plants. *International Journal of Production Research*, 46(10):2565–2588, 2008.

[7] Y. Chen, S. Lin, and S. Chou. An efficient two-staged approach for generating block layouts. *Computers & Operations Research*, 29(5):489–504, 2002.

[8] J. Chung and J. Jang. The integrated room layout for a semiconductor facility plan. *IEEE Transactions on Semiconductor Manufacturing*, 20(4):517–527, 2007.

[9] J. Chung and J.M.A. Tanchoco. The double row layout problem. *International Journal of Production Research*, 48(3):709–727, 2010.

[10] M. Delorme, M. Iori, and S. Martello. Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research*, 255(1):1–20, 2016.

[11] . Fernández, C. Gil, R. Baños, and M.G. Montoya. A parallel multi-objective algorithm for two-dimensional bin packing with rotations and load balancing. *Expert Systems with Applications*, 40(13):5169 – 5180, 2013.

[12] C.D. Geiger, R. Hase, C.G. Takoudis, and R. Uzsoy. Alternative facility layouts for semi-conductor wafer fabrication facilities. *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part C*, 20(2):152–163, 1997.

[13] J.F. Gonçalves and M.G.C. Resende. A biased random key genetic algorithm for 2D and 3D bin packing problems. *International Journal of Production Economics*, 145(2):500–510, 2013.

[14] S. Hong, D. Zhang, H.C. Lau, X. Zeng, and Y. Si. A hybrid heuristic algorithm for the 2D variable-sized bin packing problem. *European Journal of Operational Research*, 238(1):95–103, 2014.

[15] A.A. Khan, J.R. Moyne, and D.M. Tilbury. An approach for factory-wide control utilizing virtual metrology. *IEEE Transactions on Semiconductor Manufacturing*, 20(4):364–375, 2007.

[16] A. Konak, S. Kulturel-Konak, B.A. Norman, and A.E. Smith. A new mixed integer programming formulation for facility layout design using flexible bays. *Operations Research Letters*, 34(6):660–672, 2006.

[17] S. Kulturel-Konak. A linear programming embedded probabilistic tabu search for the unequal-area facility layout problem with flexible bays. *European Journal of Operational Research*, 223 (3):614–625, 2012.

[18] S. Kulturel-Konak, A.E. Smith, and B.A. Norman. Layout optimization considering production uncertainty and routing flexibility. *International Journal of Production Research*, 42(21):4475–4493, 2004.

[19] C.C. Murray, X.Q. Zuo, and A.E. Smith. An extended double row layout problem. *Proc. 12th Int. Mater. Handling Res. Colloquium*, 2012.

[20] B.A. Norman and A.E. Smith. A continuous approach to considering uncertainty in facility design. *Computers & Operations Research*, 33(6):1760–1775, 2006.

[21] B.A. Norman, R.A. Arapoglu, and A.E. Smith. Integrated facilities design using a contour distance metric. *IIE Transactions*, 33(4):337–344, 2001.

[22] G. Ozdemir, A.E. Smith, and B.A. Norman. Incorporating heterogeneous distance metrics within block layout design. *International Journal of Production Research*, 41(5):1045–1056, 2003.

[23] B.A. Peters and T. Yang. Integrated facility layout and material handling system design in semiconductor fabrication facilities. *IEEE Transactions on Semiconductor Manufacturing*, 10 (3):360–369, 1997.

[24] G.A. Susto, S. Pampuri, A. Schirru, A. Beghi, and G. De Nicolao. Multi-step virtual metrology for semiconductor manufacturing: A multilevel and regularization methods-based approach. *Computers & Operations Research*, 53:328–337, 2015.

[25] A. Taghavi and A. Murat. A heuristic procedure for the integrated facility layout design and flow assignment problem. *Computers & Industrial Engineering*, 61(1):55–63, 2011.

[26] W. Trybula, R.L. Wright, K.M. Adusumilli, and R.K. Goodall. An analysis: Traditional semiconductor lithography versus emerging technology (nano imprint). *Proceedings of the Winter Simulation Conference, 2005.*, 2005.

[27] K. Ueda, N. Fujii, I. Hatono, and M. Kobayashi. Facility layout planning using self-organization method. *CIRP Annals - Manufacturing Technology*, 51(1):399–402, 2002.

[28] L. Wei, W. Oon, W. Zhu, and A. Lim. A goal-driven approach to the 2D bin packing and variable-sized bin packing problems. *European Journal of Operational Research*, 224(1):110–121, 2013.

[29] T. Yang and B.A. Peters. A spine layout design method for semiconductor fabrication facilities containing automated material handling systems. *International Journal of Operations & Production Management*, 17(5):490–501, 1997.

[30] T. Yang, M. Rajasekharan, and B.A. Peters. Semiconductor fabrication facility design using a hybrid search methodology. *Computers & Industrial Engineering*, 36(3):565–583, 1999.

[31] T. Yang, C. Su, and Y. Hsu. Systematic layout planning: a study on semiconductor wafer fabrication facilities. *International Journal of Operations & Production Management*, 20(11): 1359–1371, 2000.

[32] T. Yang, Y. Chang, and Y. Yang. Fuzzy multiple attribute decision-making method for a large 300-mm fab layout design. *International Journal of Production Research*, 50(1):119–132, 2012.

[33] Y. Zhang, F.Y.L. Chin, H. Ting, X. Han, C.K. Poon, Y.H. Tsin, and D. Ye. Online algorithms for 1-space bounded 2-dimensional bin packing and square packing. *Theoretical Computer Science*, 554:135–149, 2014.

[34] Z. Zhang and C.C. Murray. A corrected formulation for the double row layout problem. *International Journal of Production Research*, 50(15):4220–4223, 2012.

[35] X. Zuo, C.C. Murray, and A.E. Smith. Solving an extended double row layout problem using multiobjective tabu search and linear programming. *IEEE Transactions on Automation Science and Engineering*, 11(4):1122–1132, 2014.

[36] X.Q. Zuo, C.C. Murray, and A.E. Smith. Sharing clearances to improve machine layout. *International Journal of Production Research*, 0(0):1–14, 0.

Chapter 3

Robotics in order picking: Evaluating warehouse layout for pick, place, and transport-vehicle routing systems

3.1 Introduction

It is estimated that order-picking operations can account for roughly 65% of the total operating cost, and 60% of all labor activities, in a warehouse [54]. Recent technological advances in mobile robotics promise to reduce these costs. For example, the recently-unveiled "Fetch" and "Freight" robots from Fetch Robotics, Inc., pictured in Figure 3.1 and detailed in Wise et al. [80], have been marketed to the warehousing industry to improve order picking operations. Both robots are mobile and feature onboard laser scanners to detect and avoid obstacles. The Fetch robot is equipped with a camera system to identify the items to be picked and a gripper attachment for retrieving items from a storage rack. The smaller Freight robot is designed to transport items placed within a removable tote to a packing station where the items are prepared for shipping. Freight may be used in conjunction with Fetch, with Fetch placing items into Freight for transport in a *pick-and-place* process. Alternatively, Freight may be programmed to follow a human picker in a *follow-pick* system.

This paper discusses the potential timesavings that new robotics technologies may offer in order-picking operations. In particular, we consider the problem of collecting a pick list of items. Each item on the list occupies a particular space in the warehouse (a picking location), defined by specific two-dimensional coordinates and a height on the storage rack. A "picker" robot (e.g., Fetch) must retrieve these items individually from the storage area, and a "transport" robot (e.g., Freight) must transport them to a single packing station located within the warehouse. The objective is to minimize the time required to transport all items on the pick list from the warehouse to the packing station. We term this problem

(a) Fetch and Freight

(b) Fetch's robot hand for order retrieval

Figure 3.1: Fetch and Freight (source: fetchrobotics.com).

the *pick, place, and transport vehicle routing problem* (PPT-VRP). While this problem was inspired by Fetch and Freight, it is not specific to these particular robots.

The availability of mobile picker and transport robots prompts a number of interesting research questions in the context of this order-picking problem. For example, what combination of picker and transport robots is required to obtain performance exceeding human-based picking operations, where human workers pick items and return all items described in the pick list to the packing station manually? Furthermore, how does this answer change if the robots have a constrained payload capacity (i.e., less than the size of the pick list)? To help answer these questions, a mixed integer linear programming (MILP) formulation of the PPT-VRP is proposed. Solutions to this problem describe the sequence of items to be collected by picker robots and establish the timing coordination between picker and transport robots.

This research also explores the benefits of a hybrid system in which humans are tasked to retrieve (pick) items while mobile robots are employed to transport these items to the packing station. Such a "follow-pick" system acknowledges that humans are (at least presently) more adept at identifying and grasping items from a storage shelf. It also leverages the faster travel speeds for the robotic transport unit. In particular, the maximum speed for Freight is 2.0 m/s [80]. By comparison, Roodbergen and de Koster [69] note that human pickers pushing a cart travel at about 0.6 m/s. Without a cart, a human picker may travel at 1.0 m/s [81], which is the same travel speed as the Fetch robot. The proposed PPT-VRP formulation may be employed to determine the optimal number of transport robots to pair with a given number of human pickers.

Finally, this research examines the relationships between warehouse design and mobile robot picking operations. We explore the impacts of various warehouse aspect ratios (the width of the warehouse divided by its height), the number of cross aisles within the warehouse, and the location of packing stations within the warehouse. This paper also investigates the relative impacts of altering the mix and functionality of a fleet of pick-and-transport robots. This analysis provides insight into whether it is more beneficial to add an extra picker, add another transporter, increase the carrying capacity of a transporter, or increase the retrieval speed of a picker.

The remainder of this paper is organized as follows. Related literature – including warehouse layout design and evaluation, order batching and picking, automated guided vehicle (AGV) based warehouse operations, and related vehicle routing problems – is discussed in Section 3.2. This is followed in Section 3.3 by a formal mathematical programming model of the problem. This model is extended to consider combinations of robots and human pickers, as well as an environment in which only human pickers are available. We demonstrate empirically, via an extensive numerical analysis in Section 3.4, the impacts of various warehouse layouts and highlight the relative benefits associated with modifying the robot fleet's composition and capabilities. Finally, a summary and an overview of future research opportunities are provided in Section 3.5.

3.2 Related literature

There is a vast body of warehouse operation and layout design research, recent reviews of which may be found in de Koster et al. [41], and Gu et al. [49, 50]. One category of layout research involves unit-load warehousing, where large unit sizes (e.g., pallets) limit each picker to transporting only one item at a time. Pohl et al. [66] investigated optimization of warehouse layout structures with perpendicular aisles under a dual-command operation, where workers re-stock one item and retrieve another item in a route. Gue and Meller [51] introduced novel non-traditional warehouse designs with diagonal cross aisles and non-parallel picking aisles to reduce expected traveling distances for order retrieval. Ozturkoglu et al. [60] proved that the non-traditional single cross aisle Chevron design outperforms others with more cross aisles. Other considerations of non-traditional layouts include turnover-based storage policies [67], and multiple depots [52, 61]. The common goal among these works is the minimization of the expected traveling distance (or time) by changing warehouse layouts or storage policies. Order batching and detailed picker routing were not considered in these works, as they focused on unit-loads.

Another categorization of the layout research involves batch picking, in which each picker may retrieve several items in a route. Roodbergen and Vis [71] investigated the impact of warehouse layout parameters (depot locations, number and length of aisles) and two routing policies (S-shape and largest gap) on the expected traveling distances of picking tours. Parikh and Meller [64] developed a throughput model that incorporates the vertical travel dimension for warehouses with varying lengths and heights of storage aisles. Thomas and Meller [79] discussed the effects of various layout aspects – including the size and configuration of the forward area for fast-moving-item storage, depot configuration, pallet area shape, and pallet rack height for manual case-picking warehouses – upon labor hours. Roodbergen et al. [72] provided case studies for the determination of the layout parameters that reduce the average travel distance for order picking. These layout parameters included storage unit assignments, the number of cross aisles, warehouse shape, and aisle lengths. Similarly, a simulation-based statistical analysis of certain warehouse layout parameters was conducted by Shqair et al. [77].

As stated in Gu et al. [49], proper order batching can also improve the efficiency of order retrieval. Warehouse order batching research focuses on splitting a set of orders (a pick list) into batches to ensure that all batches can be retrieved within a time window. Thus, in practice, the batch size is determined based on the distribution of the required completion time for a whole batch [65, 41]. Assuming given routing, heuristics for order batching problems have been studied under deterministic (c.f., Chen and Wu [38], Gademann and Velde [47, 48], Hsu et al. [56], Henn [53], Pan et al. [63]) and stochastic demands (c.f., Chew and Tang [39], Le-Duc and de Koster [59], Nieuwenhuyse and de Koster [103], Henn [53]). While these research works have provided efficient methodologies for order batching, they require known routing information and consider given layout configurations.

Routing methodologies for order picking also play an important role in improving warehouse efficiency. The first model for optimal picker routing was proposed by Ratliff and Rosenthal [68], which employed a traveling salesman problem to minimize order retrieval time. Improved routing models were proposed by Scholz et al. [76]. Due to the computational complexity of this problem, a number of routing heuristics for order picking – such as S-shape, largest gap, aisle-by-aisle, and combined heuristics – have been studied for problems of practical size (c.f., de Koster and Poort [40], Chew and Tang [39], Roodbergen and de Koster [69, 70], Hwang et al. [57], Theys et al. [78]).

AGV-based warehousing systems, such as implementations of autonomous vehicle based storage and retrieval system (AVS/RS) for high-density storage warehouses, have received increasing interest. For example, Ferrara et al. [45] studied warehouses with pallet shuttles and laser guided vehicles which coordinate with each other for item passing at the intersections of aisles. Queueing models were applied to estimate the order retrieval time when adjusting batch sizes of fleets. Other queueing analytic studies have addressed dwell-points (c.f., Kuo et al. [58], Roy et al. [73, 74]), aisle locations (c.f., Roy et al. [73, 74]), and batch sizes of fleets (c.f., Fukunari and Malmborg [46]). Simulation models were presented by Ekren et al. [44] to analyze the effects of system operating policies and depot locations for AVS/RS. Saidi-Mehrabad et al. [75] proposed a congestion free vehicle routing problem associated with job shop scheduling problems for AGVs transporting items from a warehouse to the manufacturing system via grid-based paths. Unlike human-based warehousing or the proposed PPT-VRP, AGV-based warehousing research assumes autonomous vehicles traveling along only designated rail guide-paths, which results in either limited warehouse zoning strategies or constrained routing policies.

In a more general context, variants of the vehicle routing problem (VRP) are also closely related to the problem at hand. Of particular relevance is the VRP with multiple synchronization constraints (VRPMS), a recent review of which is provided by Drexl [42] and efficient branch-and-cut algorithms were proposed by Drexl [43]. The VRPMS considers heterogeneous vehicles that must be coordinated to perform tasks such as load transfers or moving of truck trailers. The PPT-VRP extends the VRPMS to include queueing of delivery activities and vehicle recharging.

This paper aims to contribute to the literature in two key areas. First, the PPT-VRP represents a novel optimization problem for the coordinated routing of two types of heterogeneous vehicles (e.g., Fetches and Freights). This differs from existing routing methodologies for warehouse order retrieval, which consider routing for only individual pickers. It also removes restrictions found in AVS/RS problems in which vehicles are constrained to rail guide-paths or to particular aisles. Second, this paper explores layout design guidelines for warehouses employing picker robots or combinations of human pickers and robot transporters. The impacts of warehouse layout parameters (e.g., the number of cross aisles, the number of picking aisles, and depot locations) are examined for different vehicle parameters (e.g., vehicle quantities, speeds, and capacities).

3.3 Problem definition and formulations

The PPT-VRP may be defined as follows. A pick list (collection) of items, denoted by the set $I = \{1, \ldots, |I|\}$, must be retrieved from the warehouse and delivered to a packing station (depot). The pick list, which has been pre-defined, may contain items from multiple customer orders. The objective of the PPT-VRP is to determine routes for all available picker and transporter vehicles such that the latest time at which all items from the pick list I are dropped off at the packing station; that is, to minimize the makespan.

Two types of specialized vehicles (mobile robots) are available, where P represents the set of "picker" robots (e.g., Fetch) and D represents the set of "delivery" or "transport" robots (e.g., Freight). The entire fleet of vehicles is thus given by the set $V = P \cup D$. Note that vehicle "blocking" in aisles is ignored (i.e., traffic congestion is not considered). According to Zhang et al. [82], although there is no formal way to determine the congestion factor based on the configurations of the aisles and pickers, we checked warehouse aisle congestion studies that only considered the ratio of aisle width to picker width as from 1 [37] to 2 [55]. The ratio of aisle width to picker width in our studied warehouses is greater than 3, so we do not consider aisle congestion in this research. The configuration details of this research can be found in Section 3.4.1

Each robot $v \in V$ may have a unique payload capacity, given by $\hat{w}_v$. The initial payload carried by a given robot is denoted by w'_v, while the weight of item $i \in I$ is given by $\bar{w}_i$. Each battery-powered vehicle $v \in V$ has an initial charge of $0 \leq c'_v \leq 1$, which represents the remaining percentage of battery life. Batteries are discharged at the rate of $0 \leq d_v \leq 1$ percent per unit time, which is assumed to be independent of payload. When a vehicle visits the depot, charging stations will re-charge batteries at a rate of $0 \leq r_v \leq 1$ percent per unit time. It is assumed that a sufficient number of charging stations are available, such that vehicles do not wait for charging access. Travel time from the packing station to the chargers is assumed to be negligible.

Three types of service time are separately identified. First, the time required for picker robot $v \in P$ to grasp item $i \in I$ from its location on the stocking shelf is denoted by $s^{\text{pick}}_{v,i}$. This accounts for differences in robot capabilities as well as additional time required to grasp items located far (vertically) from the robot's default pose. Similarly, let $s^{\text{place}}_{v,i}$ represent the time required for picker robot $v \in P$ to place item $i \in I$ into a delivery robot. Finally, the time required for all items transported by delivery robot $v \in D$ to be offloaded from the robot at the depot is given by s_v^{drop}. We assume that this time is independent of the number of items held by the robot, as the person collecting these items at the depot is expected to simply replace the used tote with an empty one. As delivery vehicles arrive at the depot they form a queue while waiting for their totes to be replaced.

3.3.1 Representing the network structure

An underlying network structure facilitates the characterization of vehicle movement. This network includes three types of nodes that represent (1) the initial location of each vehicle, (2) the locations of items to be retrieved from the warehouse, and (3) the location of the depot. We let $\Delta_v^0 = 0$ represent the initial location of vehicle $v \in V$. Although this node's label equals zero for all vehicles, it is not a requirement that each vehicle actually begin service at the same physical location; this labeling convention simply serves to reduce the number of node numbers. Next, each item $i \in I$ defines a node representing the location of the item. Finally, multiple nodes are utilized to represent the single packing station (depot). Although there is one *physical* packing station, our network representation requires the creation of multiple *replicas* (copies) of this station. Each replica is given a unique number, and is associated with exactly one robot. These replicas are required because each individual robot may visit the packing station multiple times; delivery (Freight) robots may visit multiple times to deliver items or to recharge, while picker (Fetch) robots will only visit the packing station to recharge. Each time a robot visits the packing station it will be assigned to a different replica of the station. Specifically, we define Δ_v^* to be the set of packing station replicas for vehicle $v \in V$. Note that $\Delta_{v_1}^* \cap \Delta_{v_2}^* = \varnothing$ for all $v_1 \neq v_2$ (i.e., all of these nodes are unique). A pre-processing step is required to determine the number of replicas that should be created for each robot. Thus, the entire set of nodes is given by: $N = \{0\} \cup \{1, \ldots, |I|\} \cup_{v \in V} \Delta_v^*$.

Additional notation characterizes the permissable travel movements of the robots. We define Δ_v^+ to be the set of nodes to which vehicle $v \in V$ may travel, such that $\Delta_v^+ \subseteq \{I \cup \Delta_v^*\}$.

Note that $\Delta_v^0 \notin \Delta_v^+$ because a vehicle can never return to its initial location (vehicles may only *leave* the initial location). Furthermore, if an item associated with node $i \in I$ is too heavy for vehicle v, then $i \notin \Delta_v^+$. Next, given some node $j \in \Delta_v^+$ for a particular vehicle $v \in V$, $\Delta_{v,j}^-$ represents the set of nodes that could be visited immediately prior to node j. Thus, a vehicle may travel directly from node $i \in \Delta_{v,j}^-$ to node $j \in \Delta_v^+$. If $j \in I$ (i.e., if j represents the location of an item), then $\Delta_{v,j}^-$ contains Δ_v^0 (the vehicle's initial location), $I \setminus j$ (all other item locations), and Δ_v^* (all packing station replicas). However, if $j \in \Delta_v^*$ (i.e., if j is one of the packing station replicas), then $\Delta_{v,j}^-$ contains Δ_v^0 (this would mean that the vehicle travels directly from its initial location to a packing station), I (all item locations), and $\max\{\Delta_v^* < j\}$ (the largest replica node for vehicle v that is smaller than replica node j). Under this construction, a robot may move from one of its replica packing stations to another. However, it may only move to the next larger replica node. If a vehicle moves from replica to replica in an optimal solution, this is indicative of excess replicas being defined for this vehicle.

Using this notation, we define $\tau_{v,i,j}$ to be the time required for vehicle $v \in V$ to travel to node $j \in \Delta_v^+$ from node $i \in \Delta_{v,j}^-$. This parameter may include any additional travel time required by vehicles when turning corners. Note that $\Delta_v^0 = 0$ for all $v \in V$, but the travel time from Δ_v^0 to any location j, $\tau_{v,0,j}$, will incorporate the actual (potentially unique) initial location of vehicle v. Thus, $\tau_{v_1,0,j}$ does not necessarily equal $\tau_{v_2,0,j}$ for all $v_1, v_2 \in V$.

3.3.2 Decision variables

A variety of decision variables are employed in this coordinated vehicle routing problem. First, binary decision variable $x_{v,i,j}$ equals one if picker vehicle $v \in P$ travels from node $i \in \Delta_{v,j}^-$ to node $j \in \Delta_v^+$. For delivery vehicle $v \in D$, binary decision variable $y_{v,i,j}$ is similarly defined.

Coordination among the vehicles is a key component of this problem. It is assumed that items retrieved by a picker robot must be placed into a transport robot before the picker can proceed to the next item. Continuous decision variable $t_{v,j} \geq 0$ determines the time at which vehicle $v \in V$ arrives at node $j \in \Delta_v^+$ and is ready to conduct an activity at that node. For picker vehicles ($v \in P$), this time represents the earliest possible arrival to node j. For delivery vehicles ($v \in D$) receiving an item from a picker, the definition is nuanced. Here, $t_{v,j}$ represents the time at which the delivery vehicle may begin to receive item j. That is, the delivery vehicle is assumed to arrive at node $j \in I$ no earlier than the time at which the picker has actually retrieved the item. The binary decision variable $a_{v_1,v_2,j}$ establishes the pairing between a picker robot and a delivery robot at a particular item location, such that $a_{v_1,v_2,j} = 1$ if $v_1 \in P$ and $v_2 \in D$ are assigned to retrieve item $j \in I$.

Although picker and delivery vehicles are capacity constrained, only delivery vehicles may move while carrying an item. Continuous decision variable $w_{v,i,j} \geq 0$ represents the total weight of items carried by delivery vehicle $v \in D$ after leaving node j, having traveled from node i. Thus, if v travels from i to j, $w_{v,i,j}$ will include all weight loaded through location i plus the weight added at location j. Note that payload capacity limitations for picker vehicles are addressed in the definition of Δ_v^+, which prohibits a picker from visiting a location associated with an item that exceeds its capacity.

Three types of decision variables are associated with activities that occur at the depot. First, delivery vehicles form a queue when arriving at the packing station as they wait for their totes to be emptied. To monitor the order in which these vehicles arrive at the depot, binary decision variable $q_{j_1,j_2} = 1$ if vehicle $v_1 \in D$ arrives at its depot replica $j_1 \in \Delta_{v_1}^*$ before v_2 arrives at its depot replica $j_2 \in \Delta_{v_2}^*$. The battery-powered vehicles require periodic re-charging, which is performed at stations adjacent to the depot. The charge remaining on vehicle $v \in V$ when it arrives at depot replica $j \in \Delta_v^* \cup \Delta_v^0$ is given by continuous decision variable $0 \leq c_{v,j} \leq 1$. Note that the value of c_{v,Δ_v^0} is hard-coded to equal c_v' at the initial location. Third, $g_{v,j} \geq 0$ represents the duration that vehicle $v \in V$ spends charging at depot replica $j \in \Delta_v^* \cup \Delta_v^0$.

Finally, the makespan, which is to be minimized, is represented by continuous decision variable $m \geq 0$.

3.3.3 MILP formulation

The MILP formulation for the PPT-VRP is as follows.

$$
\begin{aligned}
\text{Min} \quad & m && (3.1)\\
\text{s.t.} \quad & m \geq t_{v,j} \quad \forall\, v \in D, j \in \Delta_v^*, && (3.2)\\
& \sum_{v \in P} \sum_{i \in \Delta_{v,j}^-} x_{v,i,j} = 1 \quad \forall\, j \in I, && (3.3)\\
& \sum_{v \in D} \sum_{i \in \Delta_{v,j}^-} y_{v,i,j} = 1 \quad \forall\, j \in I, && (3.4)\\
& 2a_{v_1,v_2,j} \leq \sum_{i \in \Delta_{v_1,j}^-} x_{v_1,i,j} + \sum_{i \in \Delta_{v_2,j}^-} y_{v_2,i,j} \quad \forall\, j \in I, v_1 \in P, v_2 \in D, && (3.5)\\
& a_{v_1,v_2,j} + 1 \geq \sum_{i \in \Delta_{v_1,j}^-} x_{v_1,i,j} + \sum_{i \in \Delta_{v_2,j}^-} y_{v_2,i,j} \quad \forall\, j \in I, v_1 \in P, v_2 \in D, && (3.6)\\
& \sum_{v_1 \in P} \sum_{v_2 \in D} a_{v_1,v_2,j} = 1 \quad \forall\, j \in I, && (3.7)\\
& t_{v_2,j} \geq t_{v_1,j} + s_{v_1,j}^{\text{pick}} - M(1 - a_{v_1,v_2,j}) \quad \forall\, v_1 \in P, v_2 \in D, j \in I, && (3.8)\\
& t_{v_1,j} \geq t_{v_2,i} + s_{v_1,i}^{\text{place}} + \tau_{v_1,i,j} - M(2 - a_{v_1,v_2,i} - x_{v_1,i,j}) \\
& \qquad \forall\, v_1 \in P, v_2 \in D, i \in I, j \in \{\Delta_{v_1}^+ : i \in \Delta_{v_1,j}^-\}, && (3.9)\\
& \sum_{j \in \{\Delta_v^+ : \Delta_v^0 \in \Delta_{v,j}^-\}} x_{v,\Delta_v^0,j} = 1 \quad \forall\, v \in P, && (3.10)\\
& \sum_{i \in \Delta_{v,j}^-} x_{v,i,j} = 1 \quad \forall\, v \in P, j \in \Delta_v^*, && (3.11)\\
& \sum_{i \in \Delta_{v,j}^-} x_{v,i,j} = \sum_{k \in \{\Delta_v^+ : j \in \Delta_{v,k}^-\}} x_{v,j,k} \quad \forall\, v \in P, j \in \Delta_v^+, && (3.12)
\end{aligned}
$$

$$\sum_{i \in \Delta^-_{v,j}} x_{v,i,j} \leq 1 \quad \forall\, v \in P, j \in \Delta^+_v, \tag{3.13}$$

$$\sum_{j \in \{\Delta^+_v : i \in \Delta^-_{v,j}\}} x_{v,i,j} \leq 1 \quad \forall\, v \in P, i \in \{\Delta^+_v \cup \Delta^0_v\}, \tag{3.14}$$

$$\sum_{j \in \{\Delta^+_v : \Delta^0_v \in \Delta^-_{v,j}\}} y_{v,\Delta^0_v,j} = 1 \quad \forall\, v \in D, \tag{3.15}$$

$$\sum_{i \in \Delta^-_{v,j}} y_{v,i,j} = 1 \quad \forall\, v \in D, j \in \Delta^*_v, \tag{3.16}$$

$$\sum_{i \in \Delta^-_{v,j}} y_{v,i,j} = \sum_{k \in \{\Delta^+_v : j \in \Delta^-_{v,k}\}} y_{v,j,k} \quad \forall\, v \in D, j \in \Delta^+_v, \tag{3.17}$$

$$\sum_{i \in \Delta^-_{v,j}} y_{v,i,j} \leq 1 \quad \forall\, v \in D, j \in \Delta^+_v, \tag{3.18}$$

$$\sum_{j \in \{\Delta^+_v : i \in \Delta^-_{v,j}\}} y_{v,i,j} \leq 1 \quad \forall\, v \in D, i \in \{\Delta^+_v \cup \Delta^0_v\}, \tag{3.19}$$

$$t_{v,0} = 0 \quad \forall\, v \in V, \tag{3.20}$$

$$t_{v,j} \geq t_{v,i} + \left(s^{\text{pick}}_{v,i} + s^{\text{place}}_{v,i} + \tau_{v,i,j}\right) x_{v,i,j} - M(1 - x_{v,i,j})$$

$$\forall\, v \in P, j \in \Delta^+_v, i \in \{\Delta^-_{v,j} \cap I\}, \tag{3.21}$$

$$t_{v,j} \geq t_{v,i} + \sum_{v' \in P} s^{\text{place}}_{v',i} a_{v',v,i} + \tau_{v,i,j} - M(1 - y_{v,i,j}) \quad \forall\, v \in D, j \in \Delta^+_v, i \in \{\Delta^-_{v,j} \cap I\}, \tag{3.22}$$

$$t_{v,j} \geq t_{v,i} + g_{v,i} + \tau_{v,i,j} x_{v,i,j} - M(1 - x_{v,i,j})$$

$$\forall\, v \in P, j \in \Delta^+_v, i \in \{\Delta^*_v \cup 0 : i \in \Delta^-_{v,j}\}, \tag{3.23}$$

$$t_{v,j} \geq t_{v,i} + g_{v,i} + \tau_{v,i,j} y_{v,i,j} - M(1 - y_{v,i,j})$$

$$\forall\, v \in D, j \in \Delta^+_v, i \in \{\Delta^*_v \cup 0 : i \in \Delta^-_{v,j}\}, \tag{3.24}$$

$$t_{v,j_2} \geq t_{v,j_1} + s^{\text{drop}}_v \left(\sum_{i \in \{\Delta^-_{v,j_1} \setminus (\Delta^*_v \cap j_2)\}} y_{v,i,j_1} \right) + \tau_{v,j_1,j_2} y_{v,j_1,j_2} - M(1 - y_{v,j_1,j_2})$$

$$\forall\, v \in D, j_1 \in \{\Delta^*_v \setminus \max\{\Delta^*_v\}\}, j_2 \in \{\Delta^+_v \setminus j_1\}, \tag{3.25}$$

$$w_{v,i,j} \leq \hat{w}_v y_{v,i,j} \quad \forall\, v \in D, j \in \Delta^+_v, i \in \Delta^-_{v,j}, \tag{3.26}$$

$$w_{v,\Delta_v^0,j} = (w'_v + \bar{w}_j) y_{v,\Delta_v^0,j} \quad \forall\, v \in D, j \in \{I \cap \Delta_v^+\}, \tag{3.27}$$

$$w_{v,i,j} = \bar{w}_j y_{v,i,j} \quad \forall\, v \in D, j \in I, i \in \{\Delta_{v,j}^- \cap \Delta_v^*\}, \tag{3.28}$$

$$w_{v,j,k} \geq \sum_{\substack{i \in \Delta_{v,j}^- \\ i \neq k}} w_{v,i,j} + \bar{w}_k y_{v,j,k} - \hat{w}_v (1 - y_{v,j,k}) \quad \forall\, v \in D, k \in I, j \in \{I : k \neq j\}, \tag{3.29}$$

$$g_{v,0} = 0 \quad \forall\, v \in V, \tag{3.30}$$

$$c_{v,\Delta_v^0} = c'_v \quad \forall\, v \in V, \tag{3.31}$$

$$c_{v,j} \leq c_{v,i} + r_v g_{v,i} - d_v \left(t_{v,j} - (t_{v,i} + g_{v,i}) \right)$$

$$\forall\, v \in V, j \in \Delta_v^*, i = \{\Delta_v^* \cup \Delta_v^0 : i \in \Delta_{v,j}^-\}, \tag{3.32}$$

$$t_{v_2,j_2} \geq t_{v_1,j_1} + s_{v_1}^{\text{drop}} \left(\sum_{i \in \{\Delta_{v_1,j_1}^- \setminus \Delta_{v_1}^*\}} y_{v_1,i,j_1} \right) - M(1 - q_{j_1,j_2})$$

$$\forall\, v_1 \in D, v_2 \in \{D \setminus v_1\}, j_1 \in \Delta_{v_1}^*, j_2 \in \Delta_{v_2}^*, \tag{3.33}$$

$$q_{j_1,j_2} + q_{j_2,j_1} = 1 \quad \forall\, v_1 \in D, v_2 \in \{D : v_2 > v_1\}, j_1 \in \Delta_{v_1}^*, j_2 \in \Delta_{v_2}^*, \tag{3.34}$$

$$q_{j_1,j_2} = 1 \quad \forall\, v \in D, j_1 \in \Delta_v^*, j_2 \in \{\Delta_v^* : j_2 > j_1\}, \tag{3.35}$$

$$m \geq 0, \tag{3.36}$$

$$q_{j_1,j_2} \in \{0,1\} \quad \forall\, v_1 \in D, v_2 \in \{D \setminus v_1\}, j_1 \in \Delta_{v_1}^*, j_2 \in \Delta_{v_2}^*, \tag{3.37}$$

$$t_{v,j} \geq 0 \quad \forall\, v \in V, j \in \Delta_v^+, \tag{3.38}$$

$$w_{v,i,j} \geq 0 \quad \forall\, v \in D, j \in \Delta_v^+, i \in \Delta_{v,j}^-, \tag{3.39}$$

$$y_{v,i,j} \in \{0,1\} \quad \forall\, v \in D, j \in \Delta_v^+, i \in \Delta_{v,j}^-, \tag{3.40}$$

$$x_{v,i,j} \in \{0,1\} \quad \forall\, v \in P, j \in \Delta_v^+, i \in \Delta_{v,j}^-, \tag{3.41}$$

$$a_{v_1,v_2,j} \in \{0,1\} \quad \forall\, v_1 \in P, v_2 \in D, j \in I, \tag{3.42}$$

$$0 \leq c_{v,j} \leq 1 \quad \forall\, v \in V, j \in \{\Delta_v^* \cup \Delta_v^0\}, \tag{3.43}$$

$$0 \leq g_{v,j} \quad \forall\, v \in V, j \in \Delta_v^*. \tag{3.44}$$

The objective function (3.1) seeks to minimize the latest time at which all items are delivered to the packing station (depot), as limited by Constraint (3.2). Constraints (3.3) and (3.4) ensure that each item is retrieved by a picker vehicle and placed into a delivery vehicle. Each item location must be visited by both a picker and a transporter.

Constraints (3.5)–(3.9) coordinate the picker and delivery vehicles at each item location, where Constraints (3.5), (3.6), and (3.7) set the appropriate value of $a_{v_1,v_2,j}$ to pair a picker with a transporter, while Constraints (3.8) and (3.9) establish the timing of this coordination. Conversely, Constraint (3.6) sets $a_{v_1,v_2,j} = 1$ if v_1 and v_2 meet at j. Constraint (3.7) ensures that each item is associated with exactly one picker/transporter pair. Next, Constraint (3.8) specifies that a picker may retrieve an item before a transporter arrives, but the transporter is not deemed to arrive at this location until the picker has completed the picking operation. Constraint (3.9) prohibits a picker vehicle from moving to the next location, j, until the placement of an item at i is completed.

The value of M, which represents a sufficiently large number, corresponds to an upper bound on the makespan. One valid bound may be calculated as the maximum cumulative time required to visit all nodes by a single vehicle, such that

$$M = \max\{\tau_v^{\text{upper}} + \tau_v^{\text{charging}}\}, \tag{3.45}$$

where

$$\tau_v^{\text{upper}} = \tau_{v,\Delta_v^0,\text{depot}} + \sum_{j \in I} 2\tau_{v,\text{depot},j} + \max\left\{\sum_{j \in I}\left(s_{v',j}^{\text{pick}} + s_{v',j}^{\text{place}}\right)\right\} + |I| \max\left\{s_{v''}^{\text{drop}}\right\}$$

for all $v \in V, v' \in P$, and $v'' \in D$. Here, τ_v^{upper} represents the time for vehicle $v \in V$ to travel from its initial location to the depot, then to make round-trip visits from the depot to each picking location, plus the maximum service time for the pick, place, and drop activities. The value of τ_v^{charging}, which represents the required charging time for vehicle v to travel a route of duration τ_v^{upper}, is given by

$$\tau_v^{\text{charging}} = \frac{\max\{0, (d_v \tau_v^{\text{upper}} - c'_v)\}}{r_v}.$$

Valid vehicle routes are established by Constraints (3.10)–(3.14). Constraint (3.10) requires each picker to depart from its initial location, while Constraint (3.11) ensures that each picker tour ends at a depot replica. Conservation of flow for picker vehicles is guaranteed by Constraint (3.12). Constraints (3.13) and (3.14) prohibit pickers from visiting or leaving any node more than once, respectively. Constraints (3.15)–(3.19) are analogous to Constraints (3.10)–(3.14) for delivery vehicles.

Constraints (3.20)–(3.25) incorporate travel time into the routing process, where Constraint (3.20) initializes the start time for all vehicles to be zero. Constraint (3.21) states that, if a picker travels from i to j (where i is associated with picking up an item), then the arrival time to j cannot be before the arrival time to i plus the total service time at i plus the travel time from i to j. Similarly for delivery vehicles, Constraint (3.22) guarantees that transporter's arrival time to j cannot be earlier than the summation of the arrival time to i, the placement service time performed by the partnering picker at i, and the travel time from i to j. Constraints (3.23) and (3.24) ensure valid start times when a picker or transporter leave a depot replica, respectively, while Constraint (3.25) captures the drop-off time required before visiting subsequent locations.

Payload limitations associated with delivery vehicles are addressed by Constraints (3.26)–(3.29). Constraint (3.26) states that the total weight carried by delivery vehicle $v \in D$ after visiting node j cannot exceed the capacity limit. When a transporter leaves its initial location (Δ_v^0) and travels to some location j, the total carried weight equals the summation of the initial weight and the quantity picked up at location j, as in Constraint (3.27). Constraint (3.28) determines the payload weight carried by transporter v when picking up the first item after leaving a depot replica. Constraint (3.29) forces the payload weight to be at least as large as the summation of weight when v leaves j, and the weight loaded at k.

Constraint (3.30) initializes the charging time at each vehicle's initial location to be zero. Similarly, Constraint (3.31) establishes the initial charge of vehicle v when leaving Δ_v^0. The left-hand side of Constraint (3.32) represents the charge of vehicle v when arriving at depot j. This charge cannot exceed the charge when it arrived at depot i (note that depot replicas are ordered such that i precedes j) plus the additional charge acquired while at station i minus the discharge that occurs between i and j.

Constraints (3.33), (3.34), and (3.35) address queueing of transport vehicles at the depot. Constraint (3.33) establishes the effective arrival time of a transporter at the depot, taking into account the arrival order of all transport vehicles. Decision variable $q_{j_1,j_2} = 1$ if v_1 arrives before v_2, where j_1 is the depot replica associated with v_1. The time that v_2 may begin service at the depot must not be before v_1 has completed. Constraint (3.34) considers two depot replica nodes that are used by different transport vehicles, ensuring that exactly one of the replica nodes is used before the other. Similarly, Constraint (3.35) hard-codes the values of q_{j_1,j_2} for a particular transport vehicle to force a given vehicle to utilize its replica nodes in order. The model concludes with decision variable definitions in Constraints (3.36)–(3.44).

3.3.4 Modifying the model to account for humans

While the above PPT-VRP model was formulated specifically for picker and transport robots, it is straightforward to modify the formulation to address combinations of human pickers and robotic delivery vehicles. For these mixed modes, we consider only the case of a human performing picking operations and a delivery robot transporting picked items to the depot. Given that humans are (at least presently) more adept at identifying and picking items from a shelf, and that delivery robots are likely faster at moving material, it would seem impractical to replace the delivery robot with a human. Thus, to replace the picker robot with a human (i.e., to model a follow-pick system), we consider the set P to represent all human pickers (rather than picker robots). It is then sufficient to modify the parameter values describing travel time ($\tau_{v,i,j}$), payload capacity ($\hat{w}_v$), picking time ($s^{\text{pick}}_{v,i}$), and placing time ($s^{\text{place}}_{v,i}$) for all $v \in P$. Constraints (3.30) – (3.32), which govern battery consumption, may be safely ignored for all $v \in P$.

For the purposes of comparing the robot-only and hybrid human-robot systems, it is also beneficial to determine the optimal routing assignments associated with traditional human-based order picking. In this scheme, each human worker moves through the warehouse with a cart and performs both picking and transporting operations. As in the PPT-VRP, the routing constraints prohibit any item location from being visited by more than one picker. Additionally, we assume that a queue may still be formed at the packing station. However, no charging time is required for the human. To re-use the framework of the PPT-VRP model, we let $V = P = D$ be redefined as the set of human pickers, and let decision variable $y_{v,i,j} = 1$ be redefined to indicate that human $v \in V$ should travel from location $i \in \Delta^-_{v,j}$ to location $j \in \Delta^+_v$. The human routing model presented below incorporates Constraint (3.47), which is modified from Constraint (3.22) to ensure that every human picker only leaves a picking location after performing both item picking and placing.

$$\text{Min} \quad m \tag{3.46}$$

$$\text{s.t.} \quad t_{v,j} \geq t_{v,i} + \left(s^{\text{pick}}_{v,i} + s^{\text{place}}_{v,i} + \tau_{v,i,j}\right) y_{v,i,j} - M(1 - y_{v,i,j})$$
$$\forall\, v \in D, j \in \Delta^+_v, i \in \{\Delta^-_{v,j} \cap I\}, \tag{3.47}$$
$$\text{Constraints (3.2), (3.4), (3.15)} - \text{(3.20), (3.25)} - \text{(3.29), (3.33)} - \text{(3.40).}$$

3.4 Numerical analysis

A series of numerical studies was conducted to (1) assess the impact of warehouse layout configurations on the performance of PPT-VRP systems, and (2) quantify the relative impacts of adding picker or transporter robots, increasing picker speeds, or increasing transporter capacities. All computational work was conducted on a PC with a core Intel i5-2410m processor running Microsoft Windows 8 in 64-bit mode. The PPT-VRP models were solved by Gurobi 6.0.3.

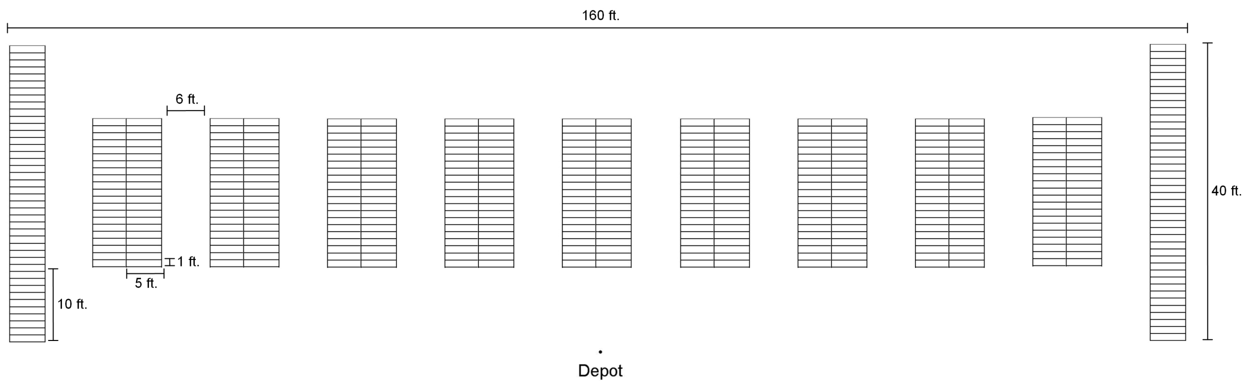
3.4.1 The impacts of layout designs on PPT-VRP

To determine the degree to which warehouse layouts impact the relative performance of robotic-based picker systems against traditional human-based picking operations, 18 different warehouse layouts were generated. Each layout is characterized by the number of vertical picking aisles (PAs), the number of horizontal cross aisles (CAs), and the location of the depot. Specifically, these test layouts feature either 2, 6, or 10 PAs; 2, 3, or 4 CAs; and either traditional or centrally-located depots. Traditional depots (TDs) are typically located at the horizontal midpoint along the lower boundary of the warehouse, while less-common central depots (CDs) are located in the middle of the warehouse.

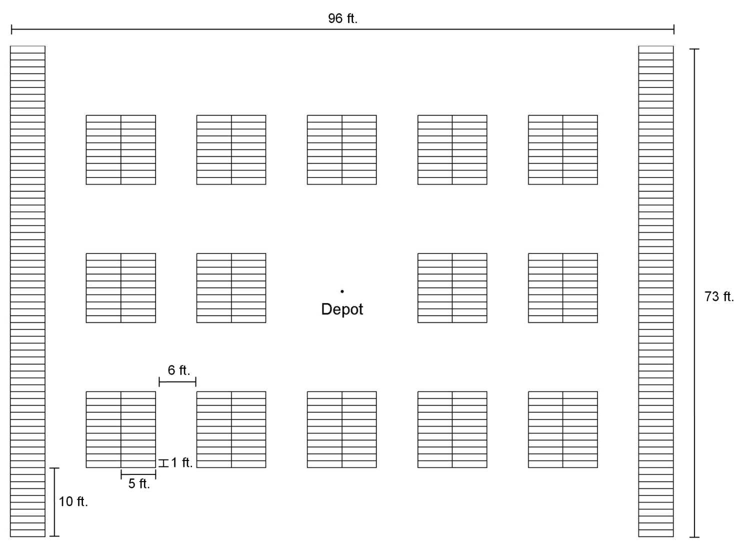
Consistent with Pan et al. [62] and Parikh and Meller [64], CAs have a width of 10 feet, PAs a width of 6 feet, and storage racks have footprints of 1-foot wide by 5-feet deep. According to the specifications of [80], both the Fetch and Freight robots have bases of 22-inches in diameter. Thus, in these layouts, three robots can occupy a PA side-by-side with room to spare. For this reason, aisle blocking is not considered in this study.

Each layout contains 460 $\pm$ 20 picking locations, with slight variations owing to the loss of picking locations surrounding CDs. Furthermore, changes in the numbers of PAs and CAs affect the quantity of storage locations. For example, as noted by [70], adding CAs increases the space requirements of a warehouse. Two of the generated warehouse layouts are illustrated in Figure 3.2.

Three metrics are employed to quantify the differences among the generated layouts. These include the aspect ratio (α), the average distance from the depot to each picking location (ADFD), and the average distance between picking locations (ADBPL). These values



(a) A warehouse with a traditional depot, 10 PAs, and 2 CAs. There are 440 storage locations in this 595-square-meter facility, identified as Layout 7 in Table 3.1.



(b) A warehouse with a central depot, 6 PAs and 4 CAs. There are 454 storage locations in this 651-square-meter facility, identified as Layout 15 in Table 3.1.

Figure 3.2: Illustrations of two generated layouts

Table 3.1: A summary of warehouse designs generated for the numerical analysis

	Layout Parameters			**Layout Attributes**				
Layout	**Depot Type**	**# of PAs**	**# of CAs**	α	**ADFD (m)**	**ADBPL (m)**	**# of Storage Locations**	**Space Req. (m^2)**
1	TD	2	2	0.26	22.82	19.90	456	368.64
2		2	3	0.25	23.74	16.65	460	386.48
3		2	4	0.24	24.66	16.53	464	404.31
4		6	2	1.78	17.48	18.50	448	481.61
5		6	3	1.52	19.10	17.09	456	561.88
6		6	4	1.32	20.71	17.62	464	642.15
7		10	2	4	20.67	23.40	440	594.58
8		10	3	3.2	22.58	22.89	460	743.22
9		10	4	2.67	22.89	23.84	480	891.87
10	CD	2	2	0.25	12.07	20.82	456	386.48
11		2	3	0.24	12.19	16.89	458	389.45
12		2	4	0.23	13.32	17.50	464	422.15
13		6	2	1.71	16.70	19.01	448	499.45
14		6	3	1.52	12.22	17.18	450	561.88
15		6	4	1.32	14.37	18.04	454	651.06
16		10	2	3.81	20.49	24.08	456	624.31
17		10	3	3.2	16.77	22.98	456	743.22
18		10	4	2.67	19.00	24.14	466	891.87

are summarized in Table 3.1, along with the total space requirements and the number of storage locations. The ADFD and ADBPL metrics have been widely used to estimate the travel distances of different warehouse picker routing policies (c.f., Hwang et al. [57], Roodbergen and Vis [71], Gue and Meller [51], Ozturkoglu et al. [60]). Note that layouts with traditional depots have a higher ADFD, but a slightly lower ADBPL, than layouts with centrally-located depots.

For each layout, 300 randomly-generated 5-item pick lists were created, resulting in 5,400 test instances. A uniform storage policy is applied, as per previous batch order picking research (c.f., Ho et al. [54], Nieuwenhuyse and de Koster [103], Parikh and Meller [64], Yu and de Koster [81], Henn [53], Shqair et al. [77], Thomas and Meller [79], Roodbergen et al. [72]). The use of fixed pick list sizes is common in the order picking literature (c.f., Roodbergen and de Koster [70], Hwang et al. [57], Roodbergen and Vis [71], Pan et al. [62], Shqair et al.

[77]). However, while these studies considered pick list sizes ranging from 4 to 80 items, the complexity of the PPT-VRP would require excessive computational time to obtain optimal solutions for larger-sized pick lists via Gurobi. Of course employing heuristic methods would provide solutions to solve larger-scale problems, but these would not be provably optimal. As such, any analysis regarding the impacts of layout configurations would be complicated by an uncertain optimality gap.

Three order-picking systems are evaluated. The first is based on the Fetch and Freight picker and transporter robots, denoted below as F&F. The second combines human pickers with robotic transports, denoted as H&F (or human and Freight). In this follow-pick collaboration, humans perform the tasks of picking and placing items, while the robotic transporter performs delivery operations. The H&F approach leverages the shorter item retrieval times for humans versus picker robots and the faster travel speeds for transporter vehicles versus humans. Finally, the third system considers traditional human-based picking.

Nine combinations of pickers and transporters are considered – from one picker and one transporter (1/1), to three pickers and three transporters (3/3) – in the F&F and H&F systems. A single human is considered in the human-only system.

Picker robots travel at a speed of 1.0 m/s, while transport robots travel at 2.0 m/s [80]. Each picker robot is assumed to require 5-seconds to pick up or place an item into a tote. Humans are assumed to travel at 0.6 m/s with a cart, and 1 m/s without; humans require 1.5 seconds to pick up and place an item into a cart [81].

The tote drop-off time is assumed to be 5 seconds for either transport robots or humans. The capacity of each transporter is varied, such that each may hold 1, 3 or 5 items in a tote. This allows cases where a transporter with relatively low capacity must revisit the depot to fulfill a pick list. No capacity limitations are placed on human-based operations.

In the following analysis, "percentage improvement" refers to the improvement in efficiency relative to that of a human-based system with a single human worker. We first investigate the impacts associated with changing the number of CAs, as summarized in Figure 3.3. Each individual boxplot was obtained by using either an F&F or H&F system and either a TD or CD. According to the results, adding CAs does not improve the performance substantially. In the cases in which single one-unit-capacity transporters were used in warehouses with TDs, the efficiency improvement decreases by an average of 0.4% per additional CA, because inserting more CAs pushes the picking locations farther away from the TD (i.e., increases the ADFD). Consequently, narrow layouts are not suitable for use in combination with low-capacity mobile systems. This finding is consistent with those of Roodbergen and de Koster [70], who noted that increasing the number of CAs might not improve the performance of human-based systems despite creating more picking route options. Thus, although adding CAs decreases the ADBPL, it also increases the ADFD. Nevertheless, this impact is reduced when a CD is adopted; CDs outperform TDs by 4.8% overall and by 7.87% when low-capacity transporters are employed.

Moreover, neither the layout parameters nor the picker type (humans or robots) noticeably impact the efficiency in the 3/3 allocation, even if low-capacity transporters are used. This result implies that if the numbers of pickers and transporters are sufficient (compared to the length of the pick list), the efficiency improvement realized by using robots becomes "robust" against changes in the layouts and type of pickers. Additionally, it is evident that the human-only picking system is more efficient (on average) than the robotic systems involving single transporter robots with one-unit capacities (e.g., the human-only system is 5.2%, 4.3%, and 7.6% more efficient than the 1/1, 2/1, and 3/1 robotic systems, respectively). The negative efficiency improvement percentages in Figure 3.3 provide evidence of this characteristic. However, a closer analysis reveals that, in situations with large ADFDs, the robotic-based systems are more efficient since transport robots can travel faster than humans.

Figure 3.4 illustrates the performance efficiency impacts associated with changing the number of PAs. Each additional PA increases the efficiency by 3.3% on average in both the

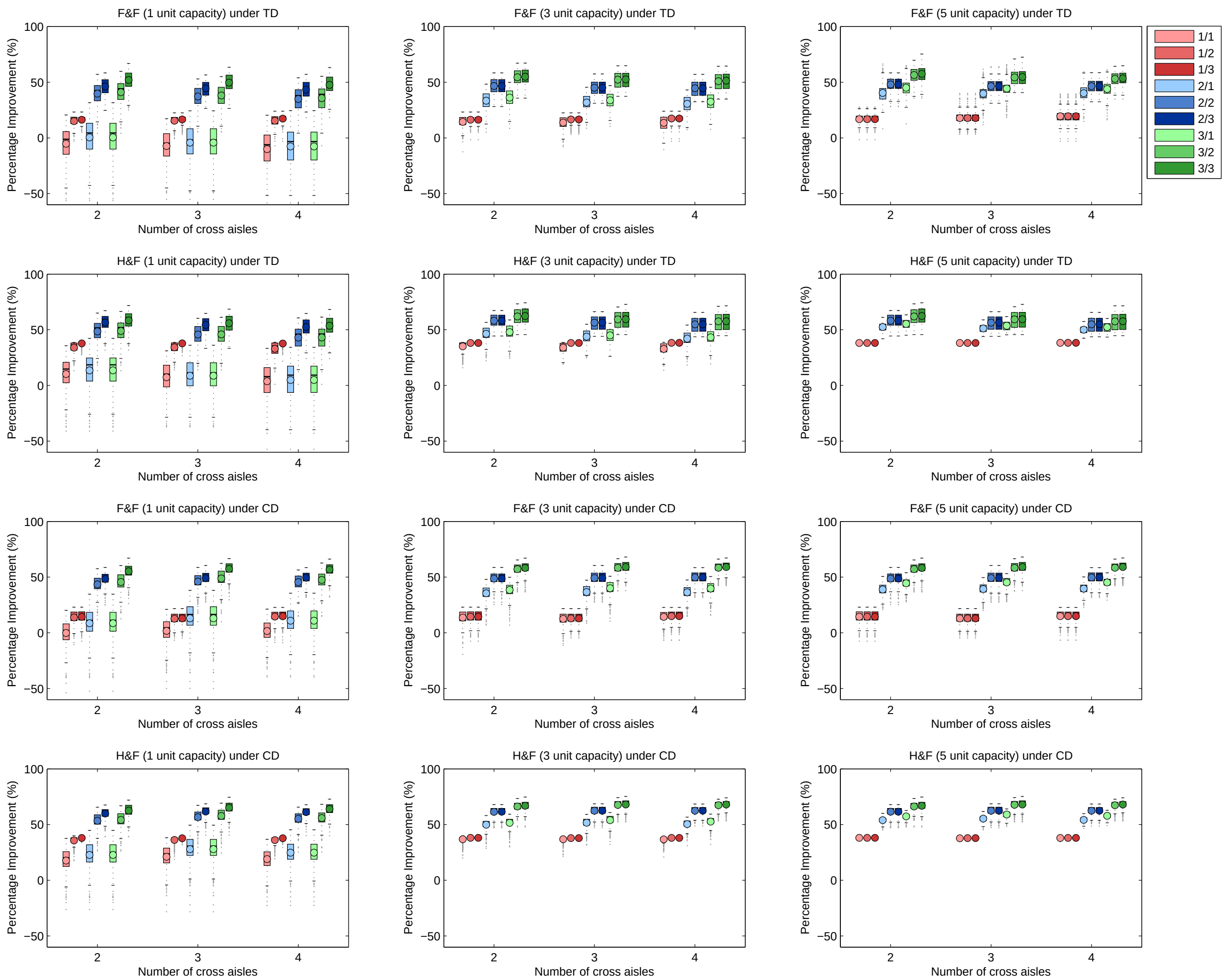


Figure 3.3: Impacts of CAs on robotic picking efficiency

situations with TDs and those with CDs. This improvement results from the fact that adding PAs increases the ADBPL, which causes the distances that must be traveled to visit the random picking locations to increase. Thus, the high speed of transporter robots becomes more advantageous as the number of PAs increases. The improvements are reduced for systems with single low-capacity transporters when the ADFD increases, but are greater in warehouses with higher ADBPLs, in which cases humans must travel for significantly greater times than robots to visit the picking locations in a route. Moreover, although implementing a CD reduces the impact of adding CAs, ADBPL is not significantly impacted by adding PAs when a CD is used. The analysis also indicates that there is no benefit associated with having more transporters than pickers when the transporter capacity is high. This finding is not surprising, because in these cases each transporter simply follows a picker during the item retrieval process, given that transporters are faster than pickers and have capacities sufficient to hold all of the items on the pick list in a route.

Due to the randomness of the generated pick lists, the ranges of the boxplots obtained using certain combinations of layout parameters appear to be rather wide. For example, in a certain warehouse, the efficiency improvement could be negative if all the items from the pick list were located close to the depot but positive if the items were located far away from each other or from the depot. Thus, we investigated the effects of the ADFD and ADBFL on the efficiency, regardless of the layout parameters. Figure 3.5 depicts the relationships among the makespan, ADFD, and ADBPL for each generated pick list. The vehicle combination used to obtain each three-dimensional plot is indicated above the plot, and the differently colored surfaces correspond to F&F and H&F systems with different capacities. Some of the three-dimensional plots have only four surfaces because the same results were obtained by using three- and five-unit-capacity transporters, in both the H&F and F&F cases.

As shown in Figure 3.5, the impact of the ADBPL is relatively small compared to that of the ADFD, but interactions between the ADFD and ADBPL exist for some combinations. The interactions appear in the plots corresponding to H&F systems in which the transporters

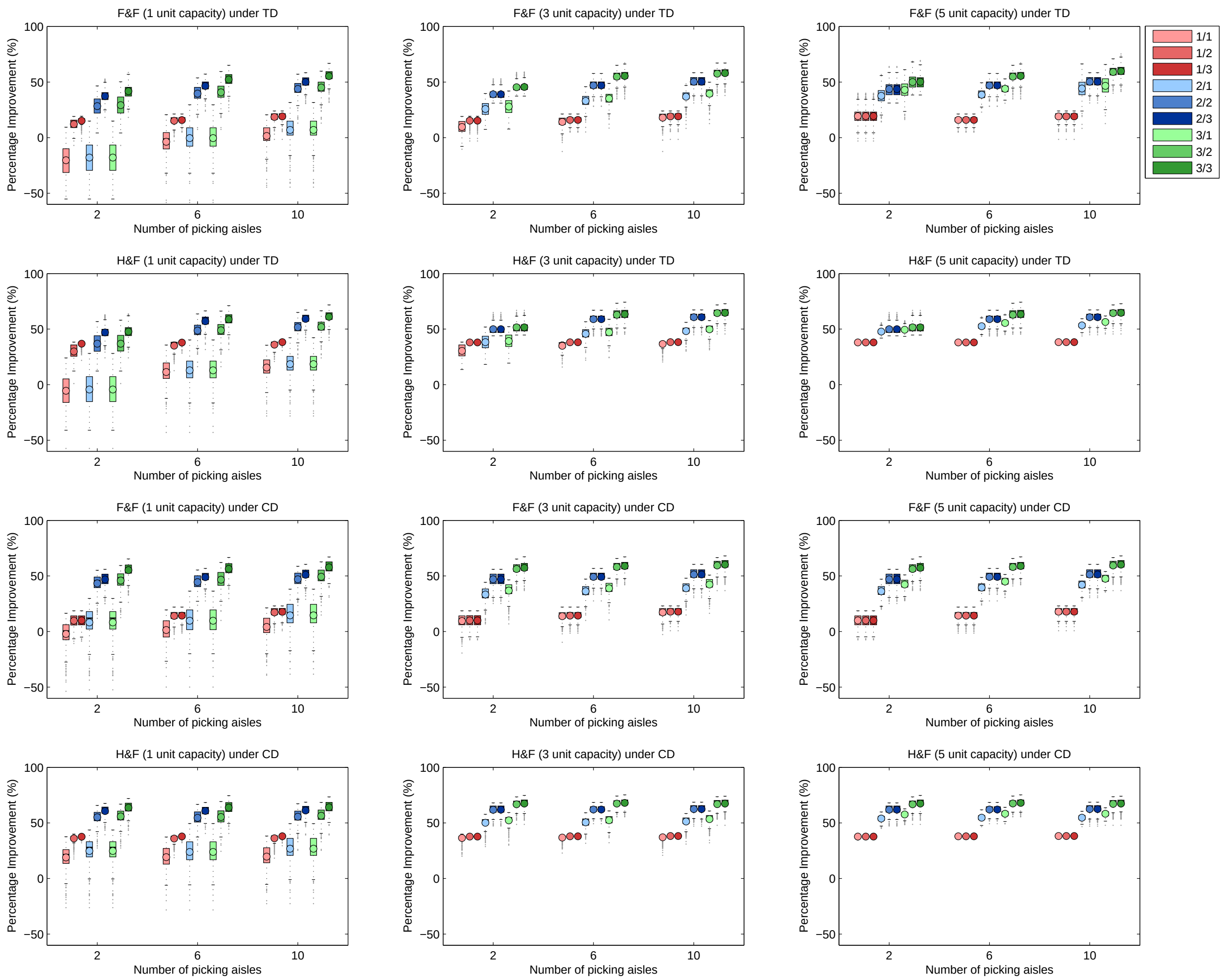


Figure 3.4: Impacts of PAs on robotic picking efficiency

have low capacities and F&F systems in which transporters have high capacities (e.g., the H&F system in which the transporters have three-unit capacities and in F&F systems with five-unit capacities). When there is only one picker and one transporter (the top left plot in Figure 3.5), the F&F system in which the transporters have five-unit capacities is more efficient than the H&F system in which they have three-unit capacities for retrieving pick lists with ADFDs greater than 23 m and ADBPLs less than 25 m. Furthermore, when there is one picker but two (three) transporters, the F&F system is more appropriate for retrieving pick lists with ADFDs greater than 15 m (12 m) and ADBPLs less than 28 m (30 m). Thus, increasing the transporters' capacities yields more substantial efficiency improvements than increasing the item picking and placing speeds does when the ADFD is relatively high. However, the improvement achieved by increasing the transporters' capacities is diminished when the ADBPL is high. The same characteristics are observable between the H&F system in which the transporters have one-unit capacities and the F&F system in which they have three-unit capacities, as shown in the plots labeled 1P/2T, 2P/3T, and 3P/3T. The trade-off of increasing the transporters' capacities is that the amount of travel among the picking locations increases, even though the frequency of travel to and from the depot decreases. Furthermore, having more pickers reduces the difference between the improvements realized by using the F&F and H&F systems (i.e., the impacts of the item picking/placing speeds), and having more transporters lessens the impacts of the transporters' capacities.

Additionally, the 2/2 systems always outperform the 3/1 systems (yielding 16.5% greater efficiency improvements, on average), which indicates the importance of balancing the numbers of pickers and transporters. For example, adding a transporter is more beneficial than adding a picker in a 2/1 system. The results also indicate that systems with fewer pickers and transporters, such as those with only one of each, are less efficient than the low-capacity F&F systems with 3/3 allocations, even if a high-capacity H&F system is employed.

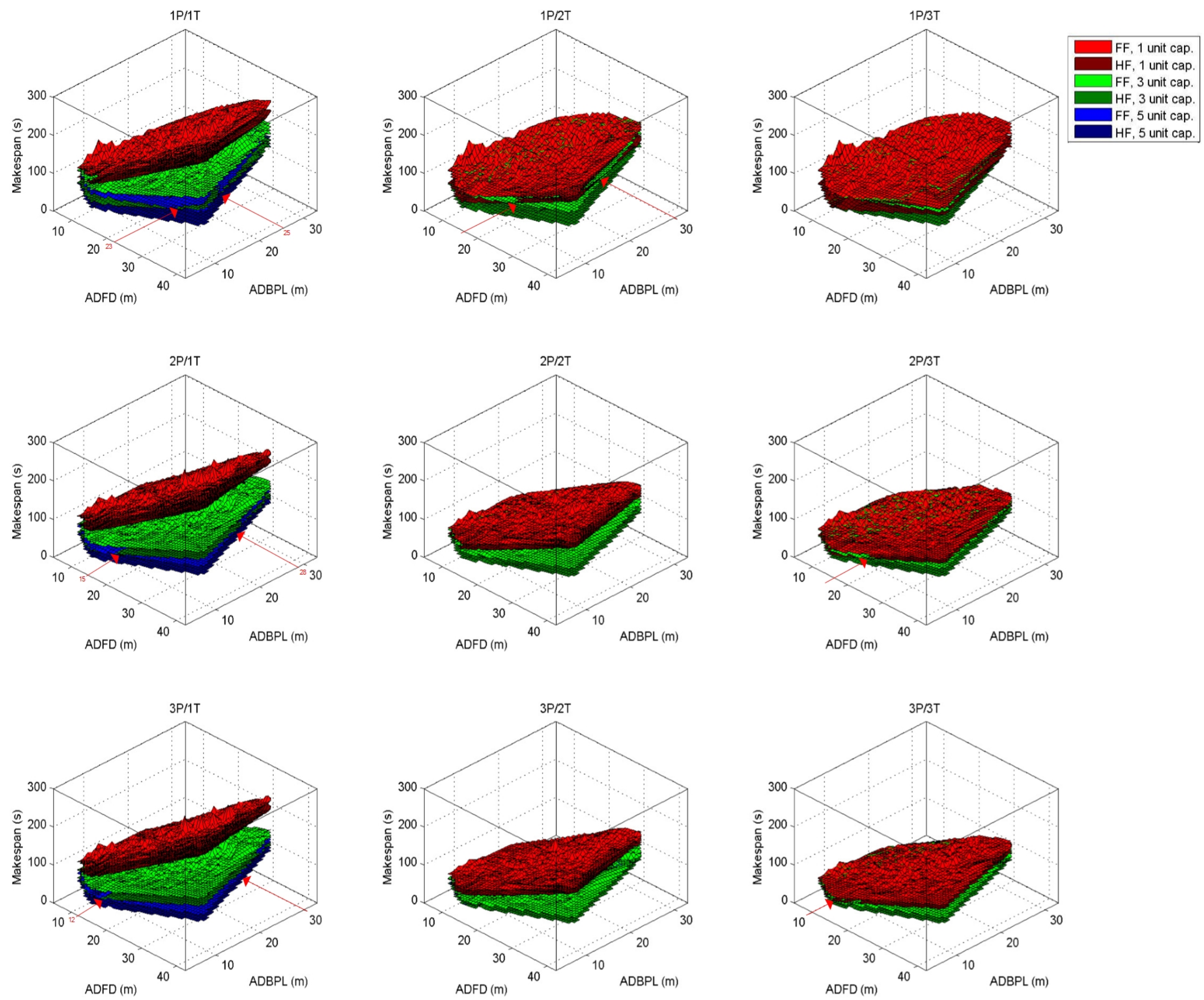


Figure 3.5: Makespans of systems with different layout attributes

3.4.2 System efficiency improvement

While the above analysis explored the impacts of layouts on a fixed assortment of order pickers (be they robots, humans, or a combination), we now turn our attention to the question of how to modify the order picker assortment. In particular, we consider four types of modifications. The first option is to add an additional picker, denoted as "+1P". Second, there is an option of adding a transporter, "+1T". Next, it may be possible to increase the capacity of a transporter. In our analysis we consider transporters that may carry one, three, or five units at a time. Thus, we denote an increase of capacity as "+2C". Finally, as technology continues to improve, it may be possible to replace the current picker robots with new models that are able to grasp and place items faster. In this analysis we consider the particular upgrade that is equivalent to replacing Fetch robots with a human picker, denoted as "F-to-H".

Consistent with the previous analysis, we limit the number of pickers and transporters to 3 or less, and the capacity of transporters to 5 or less. Thus, +1P (+1T) is not an option if 3 pickers (transporters) are already in the system and +2C is not an option for transporters that already have a capacity of 5 units. Hereafter we refer to transporters with a capacity of 1, 3, or 5 items as low (L), medium (M), or high (H), respectively. All transporters in a given system are assumed to have the same capacity. Vehicle quantities are represented as the number of pickers / number of transporters / capacity of transporters. For example, 2/1/M represents a system with 2 pickers and 1 transporter with a 3-item (medium) capacity. The details of the test problems in this numerical analysis are the same as in the previous section.

Figure 3.6 illustrates the ADFD and ADBPL combinations yielding the greatest efficiency improvements when the designated modifications were applied in situations with different transporter capacities for each pick list (regardless of the layout). The vehicle combinations used to generate the individual two-dimensional plots are provided above the plots, and colors in the plots designate the actions producing the greatest efficiency improvement percentages among all of the possible modifications and capacity levels. For example, the green area in the 1P/1T plot indicates that the efficiency improvement percentage achieved by performing +1T in the 1/1/L situation is not only greater than the improvements realized by executing any of the other three actions in the 1/1/L scenario, but also greater than those resulting from implementing any of the four modifications in the 1/1/M and 1/1/H systems.

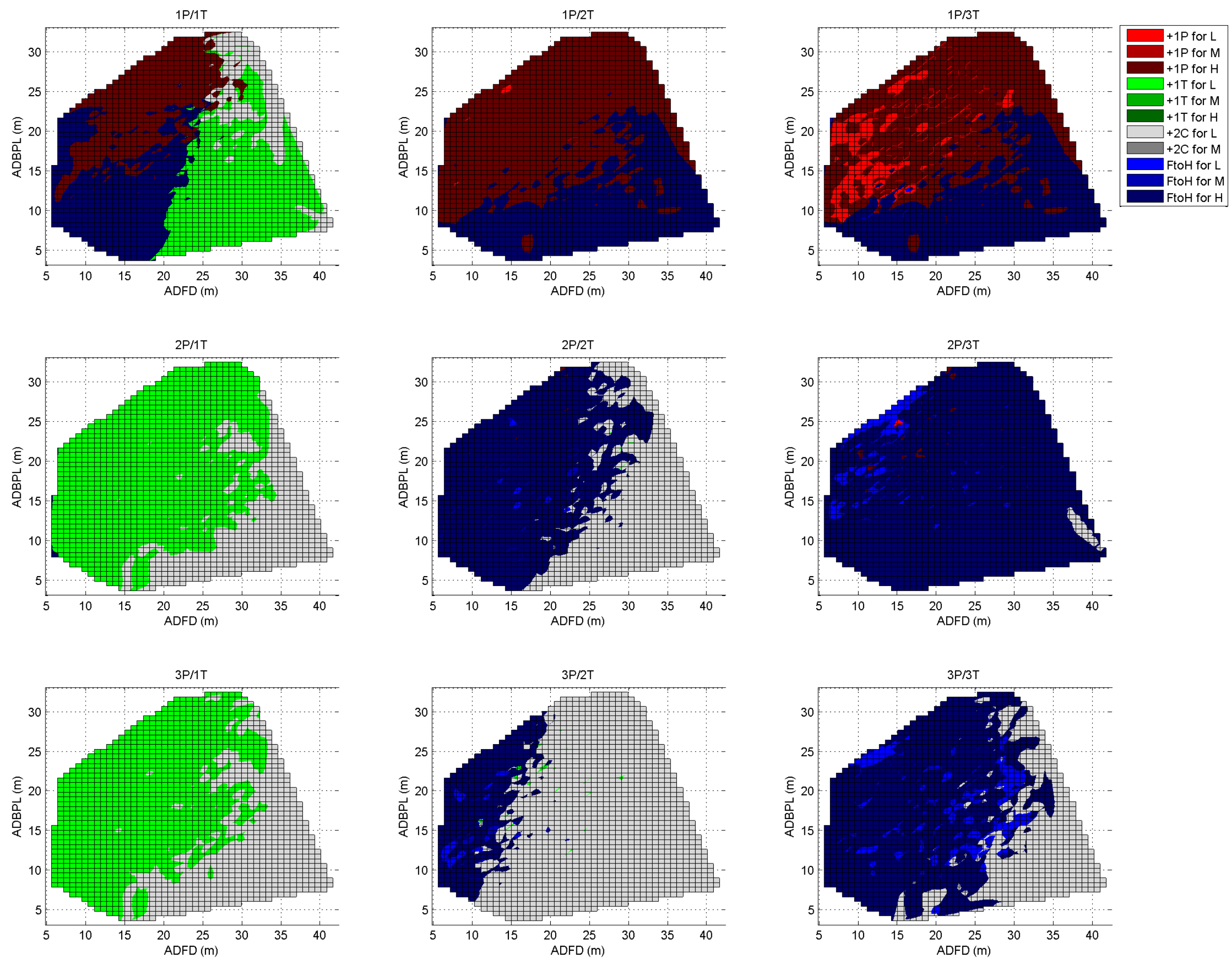


Figure 3.6: Most beneficial actions in situations with different transporter capacities and picker/transporter combinations

We categorize the vehicle combinations into four groups for further analysis. The combination in the first group, 1P/1T, has insufficient numbers of both types of vehicles (i.e., fewer vehicles than were present in the other investigated combinations). In such situations, the optimal actions depend upon the ADFD, the ADBPL, and their interaction. Note that the picker can visit all five of the picking locations in one trip but that the transporter with one-unit capacity can visit only one picking location in each trip. Thus, if the ADBPL is high, the picker must spend more time traveling among the picking locations, while if the ADFD is high, the low-capacity transporter must spend more time traveling between the depot and the picking locations. As a result, when the ADBPL is high but the ADFD is low, +1P is the optimal action in the 1P/1T scenario since the transporter can arrive at the picking locations before the picker. When the ADFD is high but the ADBPL is low, +1T is preferable. When the ADFD and ADBPL are both low, F-to-H yields the greatest improvement because not much improvement could be achieved by using robots, even though they can travel faster than humans. When the ADFD and ADBPL are both high, the transporter reaches the picking locations after the picker, since the influence of the ADFD on the transporters is greater than that of the ADBPL on the pickers.

The combinations in the second group, 1P/2T and 1P/3T, represent systems with insufficient numbers of pickers. Thus, improving the picking capabilities (by performing +1P or F-to-H) is more important than increasing the transporting capabilities (by implementing +1T or +2C). The results indicate that applying +1P yields greater efficiency improvement when the ADBPL is high since this modification reduces the distance traveled by each picker among the picking locations by dividing the route into two (by using each of two pickers to visit two or three locations rather than employing one picker to visit all five of the locations). However, F-to-H produces the greatest efficiency improvement when the ADBPL is low (in which case less improvement could be realized by applying +1P to divide the travel load).

The combinations in the third group, 2P/1T and 3P/1T, represent systems with insufficient numbers of transporters. In these cases, +2C yields the most significant efficiency improvement by reducing the frequency with which the transporters must travel between the depot and the picking locations when the ADFD is relatively high, but +1T (when using low-capacity transporters) results in greater improvement when the ADFD is low.

The combinations in the fourth group, 2P/2T, 2P/3T, 3P/2T, and 3P/3T, correspond to systems with relatively sufficient numbers of both vehicle types (i.e., more vehicles than were present in the other investigated combinations). F-to-H becomes more important than +1T when the ADFD is low, and the corresponding blue area in Figure 6 becomes larger as the number of transporters in the system increases. In addition, for the systems with low-capacity transporters, +2C only yields greater improvement than F-to-H does when the ADFD is high, but the corresponding gray area in Figure 6 shrinks as the number of transporters increases. Thus, having more transporters lessens the impact of +2C.

3.5 Conclusions and future research opportunities

This paper was motivated by the availability of specialized "picker" robots that can retrieve items from storage locations and "transport" robots that can bring these items to a packing station. Based on these capabilities, a new problem, the PPT-VRP, was defined to route these mobile robots in an effort to minimize the time required to retrieve a collection of items from within a warehouse. An MILP formulation of this problem was presented and was utilized to examine the interactions between warehouse configurations and the composition of the fleet of order-picking robots.

The numerical analysis provided several key insights. From a warehouse layout perspective, inserting more CAs pushes the picking locations farther away from the TD (i.e., increasing the ADFD that low-capacity transporters must travel to deliver each item). While employing a CD reduces the impact of adding CAs, the ADBPL is not significantly affected by adding PAs when a CD is used. From the perspective of configuring the robot fleet, in general, F-to-H is more effective when the ADFD and ADBPL are both low or there are sufficient numbers of pickers and high-capacity transporters.

When there are numerous low-capacity transporters but relatively few pickers, +1P (F-to-H) is preferable if the ADBPL is high (low). If there are numerous pickers but few low-capacity transporters, +1T (+2C) is more effective when the ADFD is low (high). When the numbers of pickers and low-capacity transporters are both sufficient, +2C (F-to-H) is preferable if the ADFD is high (low). Additionally, the impacts of the item picking and placing speeds decrease as the number of pickers increases, and the impact of the transporters' capacities decreases as the number of transporters increases. If the numbers of pickers and transporters are both sufficient, the efficiency of the system is robust against changes in the warehouse layout.

This work provides a foundation for a variety of future research opportunities. Due to the NP-hard nature of the PPT-VRP, large-scale pick lists were not considered in the analysis of optimal solutions in this study. Therefore, efficient heuristics for large-scale problems are desirable. From a warehouse operations perspective, an analysis of different storage policies would be valuable. The determination of suitable order-batching methodologies is another open topic in the context of robot-based warehousing. Another related problem of interest is the use of these robots for re-stocking activities, where transporter robots deliver items from the depot back into the warehouse and picker robots return items to the shelves.

Bibliography II

[37] F. Chen, H. Wang, C. Qi, and Y. Xie. An ant colony optimization routing algorithm for two order pickers with congestion consideration. *Computers & Industrial Engineering*, 66(1):77–85, 2013.

[38] M. Chen and H. Wu. An association-based clustering approach to order batching considering customer demand patterns. *Omega*, 33(4):333–343, 2005.

[39] E.P. Chew and L.C. Tang. Travel time analysis for general item location assignment in a rectangular warehouse. *European Journal of Operational Research*, 112(3):582–597, 1999.

[40] R. de Koster and E.V.D. Poort. Routing orderpickers in a warehouse: a comparison between optimal and heuristic solutions. *IIE Transactions*, 30(5):469–480, 1998.

[41] R. de Koster, T. Le-Duc, and K.J. Roodbergen. Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182(2):481–501, 2007.

[42] M. Drexl. Synchronization in vehicle routing: A survey of VRPs with multiple synchronization constraints. *Transportation Science*, 46(3):297–316, 2012.

[43] M. Drexl. Branch-and-cut algorithms for the vehicle routing problem with trailers and transshipments. *Networks*, 63(1):119–133, 2014.

[44] B.Y. Ekren, S.S. Heragu, A. Krishnamurthy, and C.J. Malmborg. Simulation based experimental design to identify factors affecting performance of AVS/RS. *Computers & Industrial Engineering*, 58(1):175–185, 2010.

[45] A. Ferrara, E. Gebennini, and A. Grassi. Fleet sizing of laser guided vehicles and pallet shuttles in automated warehouses. *International Journal of Production Economics*, 157:7–14, 2014.

[46] M. Fukunari and C.J. Malmborg. A network queuing approach for evaluation of performance measures in autonomous vehicle storage and retrieval systems. *European Journal of Operational Research*, 193(1):152–167, 2009.

[47] N. Gademann and V.D.S. Velde. An order batching algorithm for wave picking in a parallel-aisle warehouse. *IIE Transactions*, 33(5):385–398, 2001.

[48] N. Gademann and V.D.S. Velde. Order batching to minimize total travel time in a parallel-aisle warehouse. *IIE Transactions*, 37(1):63–75, 2005.

[49] J. Gu, M. Goetschalckx, and L.F. McGinnis. Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, 177(1):1–21, 2007.

[50] J. Gu, M. Goetschalckx, and L.F. McGinnis. Research on warehouse design and performance evaluation: A comprehensive review. *European Journal of Operational Research*, 203(3):539–549, 2010.

[51] K.R. Gue and R.D. Meller. Aisle configurations for unit-load warehouses. *IIE Transactions*, 41(3):171–182, 2009.

[52] K.R. Gue, G. Ivanovia, and R.D. Meller. A unit-load warehouse with multiple pickup and deposit points and non-traditional aisles. *Transportation Research Part E: Logistics and Transportation Review*, 48(4):795–806, 2012.

[53] S. Henn. Algorithms for on-line order batching in an order picking warehouse. *Computers & Operations Research*, 39(11):2549–2563, 2012.

[54] Y.C. Ho, T.S. Su, and Z.B. Shi. Order-batching methods for an order-picking warehouse with two cross aisles. *Computers & Industrial Engineering*, 55(2):321–347, 2008.

[55] S. Hong, A.L. Johnson, and B.A. Peters. Batch picking in narrow-aisle order picking systems with consideration for picker blocking. *European Journal of Operational Research*, 221(3): 557–570, 2012.

[56] C. Hsu, K. Chen, and M. Chen. Batching orders in warehouses by minimizing travel distance with genetic algorithms. *Computers in Industry*, 56(2):169–178, 2005.

[57] H. Hwang, Y.H. Oh, and Y.K. Lee. An evaluation of routing policies for order-picking operations in low-level picker-to-part system. *International Journal of Production Research*, 42(18): 3873–3889, 2004.

[58] P. Kuo, A. Krishnamurthy, and C.J. Malmborg. Design models for unit load storage and retrieval systems using autonomous vehicle technology and resource conserving storage and

dwell point policies. *Applied Mathematical Modelling*, 31(10):2332–2346, 2007.

[59] T. Le-Duc and R. de Koster. Travel time estimation and order batching in a 2-block warehouse. *European Journal of Operational Research*, 176(1):374–388, 2007.

[60] O. Ozturkoglu, K.R. Gue, and R.D. Meller. Optimal unit-load warehouse designs for single-command operations. *IIE Transactions*, 44(6):459–475, 2012.

[61] O. Ozturkoglu, K.R. Gue, and R.D. Meller. A constructive aisle design model for unit-load warehouses with multiple pickup and deposit points. *European Journal of Operational Research*, 236(1):382 – 394, 2014.

[62] J.C. Pan, M. Wu, and W. Chang. A travel time estimation model for a high-level picker-to-part system with class-based storage policies. *European Journal of Operational Research*, 237 (3):1054–1066, 2014.

[63] J.C. Pan, P. Shih, and M. Wu. Order batching in a pick-and-pass warehousing system with group genetic algorithm. *Omega*, 57, Part B:238–248, 2015.

[64] P.J. Parikh and R.D. Meller. A travel-time model for a person-onboard order picking system. *European Journal of Operational Research*, 200(2):385–394, 2010.

[65] C. Petersen. An evaluation of order picking policies for mail order companies. *Production and Operations Management*, 9(4):319–335, 2000.

[66] L.M. Pohl, R.D. Meller, and K.R. Gue. Optimizing fishbone aisles for dual-command operations in a warehouse. *Naval Research Logistics*, 56(5):389–403, 2009.

[67] L.M. Pohl, R.D. Meller, and K.R. Gue. Turnover-based storage in non-traditional unit-load warehouse designs. *IIE Transactions*, 43(10):703–720, 2011.

[68] H.D. Ratliff and A.S. Rosenthal. Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem. *Operations Research*, 31(3):507–521, 1983.

[69] K.J. Roodbergen and R. de Koster. Routing order pickers in a warehouse with a middle aisle. *European Journal of Operational Research*, 133(1):32–43, 2001.

[70] K.J. Roodbergen and R. de Koster. Routing methods for warehouses with multiple cross aisles. *International Journal of Production Research*, 39(9):1865–1883, 2001.

[71] K.J. Roodbergen and I.F.A. Vis. A model for warehouse layout. *IIE Transactions*, 38(10): 799–811, 2006.

[72] K.J. Roodbergen, I.F.A. Vis, and G.D. Taylor. Simultaneous determination of warehouse layout and control policies. *International Journal of Production Research*, 53(11):3306–3326, 2015.

[73] D. Roy, A. Krishnamurthy, S.S. Heragu, and C.J. Malmborg. Performance analysis and design trade-offs in warehouses with autonomous vehicle technology. *IIE Transactions*, 44(12):1045–1060, 2012.

[74] D. Roy, A. Krishnamurthy, S. Heragu, and C. Malmborg. Queuing models to analyze dwell-point and cross-aisle location in autonomous vehicle-based warehouse systems. *European Journal of Operational Research*, 242(1):72–87, 2015.

[75] M. Saidi-Mehrabad, S. Dehnavi-Arani, F. Evazabadian, and V. Mahmoodian. An ant colony algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs. *Computers & Industrial Engineering*, 86:2–13, 2015.

[76] A. Scholz, S. Henn, M. Stuhlmann, and G. Wascher. A new mathematical programming formulation for the single-picker routing problem. *European Journal of Operational Research*, 253(1):68–84, 2016.

[77] M. Shqair, S. Altarazi, and S. Al-Shihabi. A statistical study employing agent-based modeling to estimate the effects of different warehouse parameters on the distance traveled in warehouses. *Simulation Modelling Practice and Theory*, 49:122–135, 2014.

[78] C. Theys, O. Braysy, W. Dullaert, and B. Raa. Using a TSP heuristic for routing order pickers in warehouses. *European Journal of Operational Research*, 200(3):755–763, 2010.

[79] L.M. Thomas and R.D. Meller. Developing design guidelines for a case-picking warehouse. *International Journal of Production Economics*, 170, Part C:741–762, 2015.

[80] M. Wise, M. Ferguson, D. King, E. Diehr, and D. Dymesich. Fetch & Freight: Standard platforms for service robot applications. In *The IJCAI (International Joint Conference on Artificial Intelligence)-2016 Workshop on Autonomous Mobile Service Robots*, 2016.

[81] M. Yu and R. de Koster. Enhancing performance in order picking processes by dynamic storage systems. *International Journal of Production Research*, 48(16):4785–4806, 2010.

[82] M. Zhang, R. Batta, and R. Nagi. Modeling of workflow congestion and optimization of flow routing in a manufacturing/warehouse facility. *Management Science*, 55(2):267–280, 2009.

Chapter 4

Advanced pick, place, and transport-vehicle routing problems for robotics in order picking

4.1 Introduction

Recent robotics advancements promise to improve the effectiveness of order fulfillment for warehouses. In addition to autonomous warehousing, which requires operation under specific warehouse structures or configurations with rails or conveyors, a new breed of robots that are suitable to operate in normal human work environments recently became available from companies such as FetchRobotics [91], IAM Robotics [98], and Locus Robotics [101]. These robots provide not only the robustness which is able to replace (or work with) humans, but also better performance and low-cost. The operation of these solutions rely on the coordination between two types of units, pickers and transporters. A picker grasps items from a shelf and places them on a transporter. A transporter travels within a warehouse, collects items from pickers, and then quickly delivers the items to a packing station (depot) to satisfy individual customer orders.

From the operational prospective of these warehousing solutions, only Lee and Murray [100] have modeled the routing behavior for order picking as a vehicle routing problem: *the pick, place, and transport- vehicle routing problem* (PPT-VRP). The problem seeks to minimize the makespan, which is the time required to deliver all items from a pick list to the packing station. In that study, the performance of the warehousing solutions were demonstrated to be better than traditional human-based warehousing, and the impacts of warehouse layout designs was evaluated when coordinated mobile robots are deployed.

However, Lee and Murray [100] assumed that pickers cannot move while holding an item, which meant that every storage location in the pick list has to be visited by a transporter. Subsequently, we extend the study by relaxing the assumption. If a picker can move toward the transporter while holding an item picked from the storage rack, the travel distances of transporters could then be reduced by collecting items at some point between the depot and the storage rack. Thus, an advanced PPT-VRP model (APPT-VRP) that considers this functionality is proposed in this work. As illustrated in Figure 4.1, although there are two pickers and one transporter in both of these warehouses, the transporter actually travels less to retrieve two items stored in the same locations when using the APPT-VRP model.

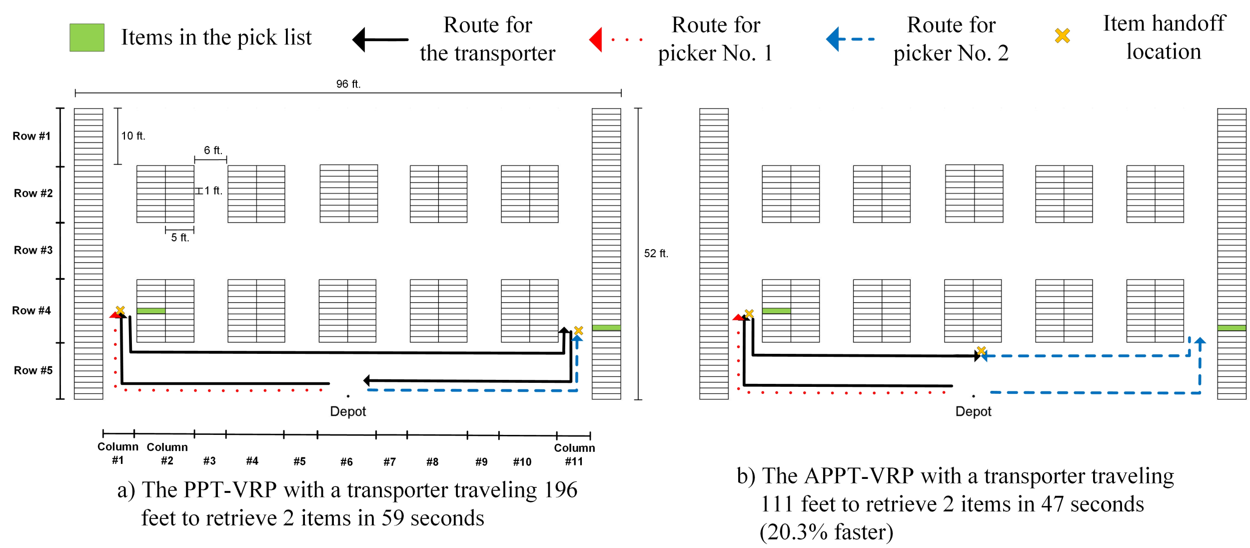


Figure 4.1: Comparison between PPT-VRP and the advanced PPT-VRP

As the PPT-VRP model, the objective of the APPT-VRP is to minimize the makespan. In Lee and Murray [100], the NP-hard nature of the PPT-VRP, the uniqueness of the PPT-VRP in the literature. And also, a typical warehouse receives thousands of orders in a day, and these orders need to be fulfilled as soon as possible for better service level. In our preliminary experimental results, a 10-item pick list problems can not be solved optimally within 24 hours. Thus, there is a need of efficient heuristics for the large-scale problems. Moreover, because of the nature of the industry, which is characterized by frequent orders from different customers that need to be satisfied as soon as possible, optimal or near-optimal routing is essential to minimize the delivery time. Therefore, this paper focuses on the development of heuristics and the demonstration of their effectiveness.

The remainder of this paper is organized as follows. A review of the literature related to autonomous vehicle routing is presented in Section 4.2. A formal problem definition of the APPT-VRP and mathematical programming model are provided in Section 4.3. As a result of the NP-hard nature of this problem, an efficient heuristic is proposed in Section 4.4 to solve large-scale APPT-VRP instances for practical use. We demonstrate the effectiveness of the proposed heuristic, and highlight the benefits associated with the APPT-VRP model via an extensive numerical analysis in Section 4.5. Finally, a summary and an overview of related future research opportunities are given in Section 4.6.

4.2 Related literature

As explained in Lee and Murray [100], the PPT-VRP model addresses the situation in which multiple heterogeneous vehicles operate collaboratively and simultaneously for warehousing. We also discussed the uniqueness of our previous study considering the collaboration between heterogeneous vehicles. Therefore, the APPT-VRP is novel and can contribute to the warehouse order picking literature. In addition to our previous survey, this section focuses on related routing problems and the literature of autonomous vehicle applications.

From the routing prospective, the first modified traveling salesperson problem (TSP) routing model, which considered the routing for single human picker warehousing, was proposed by [68], and subsequently improved by [76]. Another recently raised categorization of a related problem involves VRP with coordination constraints. Drexl [87] provides a survey article of VRP with multiple synchronization constraints (VRPMSs), and Drexl [88] proposed a branch-and-cut algorithm for VRPMSs based on a new network representation. Although VRPMS considers that autonomous fleets have to arrive at certain locations to drag compatible non-autonomous fleets with loads in order to fulfill tasks such as visiting customers and transshipment locations, routing for the non-autonomous fleets and collaboration between autonomous fleets are not considered. Other variants of VRPMS have been applied in health care services. Bredstrom and Ronnqvist [83] considered a combined vehicle routing and scheduling problem with time windows and simultaneous (several vehicles required by one customer) and precedence (multiple ordered visits required by one customer) synchronization constraints in separate models for home care service. Rousseau et al. [111] considered dispatching problems under the dynamic VRPMS. Redjem and Marcon [106], Haddadene et al. [93] proposed heuristic approaches for variants of the caregiver routing problem.

As regards VRP applications in autonomous vehicles, Murray and Chu [102] discussed a flying sidekick traveling salesman problem for drone-assisted parcel delivery. Hu et al. [96] considered the optimal routing for a multi-capacity rail guided vehicle working on a linear track in an automated freight handling system to minimize under conflict-avoidance. Davis [85] investigated the optimal manner of the dynamic routing by simulation to avoid congestion of autonomous vehicles traveling on a square lattice of roads for better vehicle safety and traffic flow. For the applications of autonomous vehicle based storage and retrieval system (AVS/RS), which are applied in high-density storage warehouses, queuing and simulation models have been proposed to analyze the effects of different operation policies (Kuo et al. [99], Fukunari and Malmborg [92], Ekren et al. [89], Roy et al. [112], Ferrara et al. [90], Roy et al. [113]). Most of these AVS/RS studies considered autonomous vehicles only traveling along rail guide-paths, with vehicles designated to particular aisles. Saidi-Mehrabad et al. [114] studied a vehicle routing problem with considering job shop scheduling for AGVs transporting for the transportation with grid-based paths between the manufacturing system and the warehouse.

Our present contribution to the literature is, as an extension of our previous study, a new PPT-VRP optimization model as an application of VRPMS in robotics warehousing. Both our present and previous studies [100] not only differ from existing routing methodologies for warehouse order retrieval, which consider routing for only individual pickers, but also remove restrictions found in AVS/RS problems in which vehicles are constrained to rail guide-paths or to particular aisles. Unlike Lee and Murray [100], the APPT-VRP enables collaboration between heterogeneous vehicles to happen anywhere in the warehouses for potential travel distance minimization by the transporting vehicles. Thus, the APPT-VRP determines the visited locations for the collaborations as well as the visiting orders for vehicles. Therefore, the APPT-VRP is novel and can contribute to the industry.

4.3 Problem definition and mathematical programming formulations

As described in [100], we set $I = \{1, \ldots, |I|\}$ as a pick list (collection) of items required to be retrieved from the warehouse and delivered to a packing station (depot). A pick list can contain only one or multiple customer orders required to be fulfilled by the vehicles. Next, let P represent the set of "picker" vehicle, and D represent the set of "delivery transporter" vehicle. Thus, the entire fleet of vehicles can be denoted by the set $V = P \cup D$, where $P \cap D = \varnothing$. Moreover, we ignore the vehicle blocking in aisles (aisle congestion) in this problem because of the size of the vehicles in practice. Additionally, pickers can not deliver the items to the depot directly without a transporter. Furthermore, robot $v \in V$ can have different initial payload capacities, denoted by w'_v, and overall payload capacities, given by $\hat{w}_v$, while the weight of item $i \in I$ is denoted by $\bar{w}_i$. As the PPT-VRP, we let $s^{\text{pick}}_{v,i}$ represent the time required for picker $v \in P$ to grasp item $i \in I$ from its location on the stocking shelf, and the time required for picker $v \in P$ to place item $i \in I$ into a transporter is denoted by $s^{\text{place}}_{v,i}$. Finally, the time required for transporter $v \in D$ to drop off (offload) a tote of items at the depot is given by s^{drop}_v.

In addition to the parameters in the PPT-VRP, we define the set of transshipment locations (item hand-off location) as $I^{\mathcal{S}} = \{|I|+1, \ldots, |I|+|I|\}$. Each element (hand-off location) in $I^{\mathcal{S}}$ corresponds to each element in I. For example, $|I|+1$ in $I^{\mathcal{S}}$ represents the item hand-off location for item 1 in I, and so as $|I|+1, \ldots, |I|+|I|$ to item $2, \ldots, |I|$, respectively. In order to address the physical coordinates of $I^{\mathcal{S}}$ in continuous space in the model, we partition the warehouses with perpendicular aisles into columns and rows as a grid-based structure. Figure 4.1 illustrates the rows and columns in a warehouse on its left side. Let c represent the c^{th} column from left to right, where $c \in C = \{1, \ldots, |C|\}$, and r represent the r^{th} row from top to bottom, where $r \in R = \{1, \ldots, |R|\}$. Then C' represents the set of rack columns in even numbers of $c \in C' = \{2, 4, \ldots\}$, and R' represents the set of cross aisle rows for all odd numbers of $r \in R' = \{1, 3, \ldots, |R|\}$. Thus, for a warehouse with the size of H on height and W on width, $H^{\mathcal{R}}_r$ represents the height of row r, and $W^{\mathcal{C}}_c$ represents the width of column c. With the x coordinate of the middle point of column c, $X^{\mathcal{C},M}_c$, and the y coordinate of the middle point of row r, $Y^{\mathcal{R},M}_r$, we can ensure item hand-off locations can only happen in aisles.

Although the distance between two known locations i and $j \in N \backslash I^{\mathcal{S}}$ can be predetermined, some notations are required to ascertain the shortest distance between two item hand-off locations or one known location (either the depot or item storage locations) and a hand-off location. Let $X^{\mathcal{K}}_i$ ($Y^{\mathcal{K}}_i$) represent x (y) coordinates of a known location, then shortest distance between two locations could be determined as the sum of the absolute differences of the coordinates if the two locations are not in the same rack column (row). Thus, additional notations are required to identify if the two locations are whether in the same rack column (row). Let $B^{\mathcal{C}}_{c,i} = 1$ if location i is in column c, where $c \in C$ and $i \in N \backslash I^{\mathcal{S}}$, and $B^{\mathcal{R}}_{r,i} = 1$ if location i is in row r, where $r \in R$ and $i \in N \backslash I^{\mathcal{S}}$. These help to identify whether two locations are in the same row (column) in order to determine the shortest distance between these two. For example, if two locations are in the same row (not cross aisles) but not in the same column, then the distance between these two locations is determined by adding a detour distance, which will be introduced later. In addition, let F_v represent the speed of vehicle v, then the required traveling time between two locations can be represented as the distance between them divided by the speed. Unlike the previous research, depot queuing is not considered in the model, which means multiple transporters can drop off items at the depot simultaneously.

4.3.1 Representing the network structure

To facilitate the characterization of vehicle movement, we extend the network structure from the PPT-VRP by incorporating four types of nodes to represent (1) the initial location of each vehicle, (2) the location of the depot, (3) the storage locations (picking locations) of items to be retrieved from the warehouse, and (4) the transshipment locations where each item is passed from a picker to a transporter in the warehouse. Note that the PPT-VRP only has (1)–(3).

In Lee and Murray [100], the initial location of vehicle $v \in V$ is denoted by $\Delta_v^0 = 0$, which allows each vehicle to begin service at different physical locations although the label of the initial location for all vehicles equals zero. Also, multiple nodes are utilized to represent the replicas of the depot for transporters, and each replica is encoded with a unique number to be associated with only one transporter. This is because each transporter may visit the depot more than one time to deliver items, so a transporter will visit a different replica each time the transporter visits the depot. The set of depot replicas for vehicle $v \in D$ is denoted by Δ_v^*, where $\Delta_{v_1}^* \cap \Delta_{v_2}^* = \varnothing$ for all $v_1 \neq v_2$, and the set of nodes is denoted by $N = \{0\} \cup I \cup I^{\mathcal{S}} \cup_{v \in V} \Delta_v^*$.

Next, item $i \in I$ ($i \in I^{\mathcal{S}}$) represents the node of the item storage (hand-off) location. Similar to [100], additional notations are employed to characterize the movement of robots. Let Δ_v^+ be the set of nodes to which vehicle $v \in V$ may travel, such that $\Delta_v^+ \subseteq \{I \cup I^{\mathcal{S}}\}\ \forall\ v \in P$ and $\Delta_v^+ \subseteq \{I^{\mathcal{S}} \cup \Delta_v^*\}\ \forall\ v \in D$, where $\Delta_v^0 \notin \Delta_v^+$ because a vehicle will never return to its initial location (vehicles only *leave* the initial location).

We let $\Delta_{v,j}^-$ represent the set of nodes that could be visited right after visiting node j. Thus, a vehicle can only travel directly from node $i \in \Delta_{v,j}^-$ to node $j \in \Delta_v^+$. If $j \in I$ (i.e., if j represents the location of an item), then $\Delta_{v,j}^-$ contains Δ_v^0 (the vehicle's initial location), $I^{\mathcal{S}} \backslash \{j + |I|\}$ (all item hand-off locations except the corresponding hand-off location of j), and Δ_v^* (all of the depot replicas) for all $v \in P$. For a transporter, if $j \in \Delta_v^*$ (depot replicas), then $\Delta_{v,j}^-$ contains Δ_v^0 (the transporter travels directly from the initial location to depot), $I^{\mathcal{S}}$

(all of the item hand-off locations), and $\max\{\Delta_v^* < j\}$ (the largest replica node for vehicle v that is smaller than replica node j). If $j \in I^{\mathcal{S}}$, $\Delta_{v,j}^-$ contains Δ_v^0, $I^{\mathcal{S}} \backslash j$, and $\Delta_v^* \backslash \max\{\Delta_v^*\}$. Under this construction, a transporter may move from one of its depot replicas to another larger replica node. Finally, we also define $\tau_{v,i,j}$ to be the required time for vehicle $v \in V$ to travel to node $j \in \Delta_v^+$ from node $i \in \Delta_{v,j}^-$ if $i, j \in N \backslash I^{\mathcal{S}}$.

Unlike Lee and Murray [100], pickers do not visit the depot, and battery charging is not considered in this paper due to the recently released specification [117] which has indicated the battery power is sufficient for the vehicles to work an 8-hour day. In such case, pickers do not have to go back to the depot after a pick list is retrieved for the purpose of continuous warehouse operation. Also, transporters can not visit any $i \in I$, although node $i \in I$ and $j \in I^{\mathcal{S}}$ can be representing the same physical location. Note that [100] assumes the hand-offs have to happen at the very item storage locations. Moreover, a picker can travel from an $j \in I$ to only the corresponding $j \in I^{\mathcal{S}}$ (an item must be passed once it is picked by a picker) but can travel from a $i \in I^{\mathcal{S}}$ to either a $j \in I \backslash \{i - |I|\}$ or to the depot.

Table 4.1 summarizes the parameter notations of APPT-VRP.

4.3.2 Decision variables

Decision variables are defined to model the APPT-VRP. As in [100], we define binary decision variable $x_{v,i,j} = 1$ ($y_{v,i,j} = 1$) if picker $v \in P$ (transporter $v \in D$) travels from node $i \in \Delta_{v,j}^-$ to node $j \in \Delta_v^+$. Decision variable $t_{v,j} \geq 0$ determines the arrival time at node $j \in \Delta_v^+$ (ready to conduct an activity at the node) for which vehicle $v \in V$. Note that if an item hand-off location is the same as the item storage location, a transporter may physically arrive at the hand-off location earlier than $t_{v,j}$ but needs to wait if the picker has not completed the picking activity. In such case, $t_{v,j}$ represents the time when the transporter begins to receive the item. Moreover, $a_{v_1,v_2,i} = 1$ if picker $v_1 \in P$ passes item $i \in I$ to transporter $v_2 \in D$ at a particular location $j = i + |I| \in I^{\mathcal{S}}$ after the item is picked

Table 4.1: Summary of parameter notations

$I = \{1, \dots, \lvert I\rvert\}$	The pick list of items required to be retrieved.
$I^{\mathcal{S}} = \{\lvert I\rvert + 1, \dots, \lvert I\rvert + \lvert I\rvert\}$	The set of item hand-off locations.
P	The set of pickers.
D	The set of transporters, where $P \cap D = \varnothing$.
$V = P \cup D$	The set of all fleets.
$C = \{1, \dots, \lvert C\rvert\}$	The set of columns in the warehouse structure, and $C' = \{2, 4, \dots\}$ represents the set of rack columns in even numbers.
$R = \{1, \dots, \lvert R\rvert\}$	The set of rows in the warehouse structure, and $R' = \{1, 3, \dots, \lvert R\rvert\}$ represents the set of cross aisle rows (odd number of rows).
H (W)	The height (width) of the warehouse.
$H^{\mathcal{R}}_r$	The height of row r.
$W^{\mathcal{C}}_c$	The width of column c.
$B^{\mathcal{C}}_{c,i} = 1$	If location i is in column c, where $c \in C$ and $i \in N \backslash I^{\mathcal{S}}$. Otherwise, $B^{\mathcal{C}}_{c,i} = 0$.
$B^{\mathcal{R}}_{r,i} = 1$	If location i is in row r, where $r \in R$ and $i \in N \backslash I^{\mathcal{S}}$. Otherwise, $B^{\mathcal{R}}_{r,i} = 0$.
$X^{\mathcal{C},M}_c$	The x coordinate of the middle point of column $c \in \mathcal{C}$.
$Y^{\mathcal{R},M}_r$	The y coordinate of the middle point of row $r \in \mathcal{R}$.
$X^{\mathcal{K}}_i$ $(Y^{\mathcal{K}}_i)$	The x (y) coordinate of a known location (the depot or an item storage location).
$\Delta^0_v = 0$	The initial location of vehicle $v \in V$.
Δ^*_v	The set of depot replicas for vehicle $v \in D$, where $\Delta^*_{v_1} \cap \Delta^*_{v_2} = \varnothing$ for all $v_1 \neq v_2$.
Δ^+_v	The set of nodes to which vehicle $v \in V$ may travel, such that $\Delta^+_v \subseteq \{I \cup I^{\mathcal{S}}\}$ $\forall$ $v \in P$ and $\Delta^+_v \subseteq \{I^{\mathcal{S}} \cup \Delta^*_v\}$ $\forall$ $v \in D$, where $\Delta^0_v \notin \Delta^+_v$.
$\Delta^-_{v,j}$	The set of nodes that could be visited right after visiting node j, where $\Delta^-_{v,j\in I} \subseteq \{\Delta^0_v \cup I^{\mathcal{S}} \backslash \{j + \lvert I\rvert\} \cup \Delta^*_v\}$ $\forall$ $v \in P$, $\Delta^-_{v,j\in\Delta^*_v} \subseteq \{\Delta^0_v \cup I^{\mathcal{S}} \cup \max\{\Delta^*_v < j\}\}$, and $\Delta^-_{v,j\in I^{\mathcal{S}}} \subseteq \{\Delta^0_v \cup I^{\mathcal{S}} \backslash j \cup \Delta^*_v \backslash \max\{\Delta^*_v\}\}$ $\forall$ $v \in D$.
$N = \{0\} \cup I \cup I^{\mathcal{S}} \cup_{v\in V} \Delta^*_v$	The set of nodes in the network structure.
F_v	The speed of vehicle $v \in V$.
$s^{\text{pick}}_{v,i}$	The time required for picker $v \in P$ to grasp item $i \in I$ from its storage location.
$s^{\text{place}}_{v,i}$	The time required for picker $v \in P$ to place item $i \in I$ into a transporter.
s^{drop}_v	The time required for transporter $v \in D$ to drop off a tote of items at the depot.
$\tau_{v,i,j}$	The required time for vehicle $v \in V$ to travel to node $j \in \Delta^+_v$ from node $i \in \Delta^-_{v,j}$ if $i, j \in N \backslash I^{\mathcal{S}}$.
w'_v	Initial payload capacity of vehicle $v \in V$.
$\hat{w}_v$	Overall payload capacity of vehicle $v \in V$.
$\bar{w}_i$	The weight of item $i \in I$.

up at location i by v_1. Additionally, continuous decision variable $w_{v,i,j} \geq 0$ represents the total weight of items carried by transporter $v \in D$ after leaving node j, having traveled from node i.

In addition to Lee and Murray [100], several types of item hand-off location associated decision variables are required. First, let $x_i^{\mathcal{S}}$ ($y_i^{\mathcal{S}}$) represent the x (y) coordinate of the item hand-off location for item i, where $i \in I^{\mathcal{S}}$, and let $d_{i,j}^X$ ($d_{i,j}^Y$) represent the x (y) distance from i to j, where $i \in N$, $j \in \{N : i \in I^{\mathcal{S}} | I^{\mathcal{S}} : i \in N \backslash I^{\mathcal{S}}\}$, and $i \neq j$. Note that this research considers the routing problem within the warehouses with perpendicular aisles, where the shortest distance between two locations may be longer than the difference between the x coordinates plus the difference between the y coordinates if these two locations are in the same rack column/row but not in the same row (column). In order to ascertain the shortest distance, $d_{i,j}^X$ and $d_{i,j}^Y$, additional variables are required to check if two locations are in the same rack column (row). Let $b_{c,i}^{\mathcal{C}} = 1$ if item $i - |I|$ is passed within column c, where $c \in C$ and $i \in I^{\mathcal{S}}$, and $b_{r,i}^{\mathcal{R}} = 1$ if item $i - |I|$ is passed within row r, where $i \in I^{\mathcal{S}}$ and $r \in R$. Then $b_{i,j}^{\mathcal{C},\text{Same}} = 1$ ($b_{i,j}^{\mathcal{R},\text{Same}} = 1$) can be determined if node i and node j are located in the same column $c \in C'$ or row $r \in R \backslash R'$. Thus, let $b_{i,j}^{\text{detour},X} = 1$ ($b_{i,j}^{\text{detour},Y} = 1$) if the distance between node i and node j is longer than the the difference between the x (y) coordinates . In other words, a horizontal (vertical) detour is required traveling from one location to the other when rack column (row) is on the way as obstacles. For horizontal detours, which mean two possible routes, the left one and the right one, on detour, let $a_{i,j}^L = 1$ ($a_{i,j}^R = 1$) if vehicle travels through the left (right) route from i to j when $b_{i,j}^{\text{detour},X} = 1$, where $i \in N$, $j \in \{N : i \in I^{\mathcal{S}} | I^{\mathcal{S}} : i \in N \backslash I^{\mathcal{S}}\}$, and $i \neq j$. For vertical detours, let $a_{i,j}^U = 1$ ($a_{i,j}^D = 1$) if vehicles travel through up/down route on detour from i to j when $b_{i,j}^{\text{detour},Y} = 1$, where $i \in N$, $j \in \{N : i \in I^{\mathcal{S}} | I^{\mathcal{S}} : i \in N \backslash I^{\mathcal{S}}\}$, and $i \neq j$. Detour routes are illustrated in Figure 4.2.

Moreover, the formulations for determining the distance between two item hand-off locations are not linear. Therefore, variables to linearize the formulations are introduced. Let $l_{i,j}^{P,L} = a_{i,j}^U \sum_{c \in C'} b_{c,j}^C X_{c-1}^{\mathcal{C},M}$ and $l_{i,j}^{P,R} = a_{i,j}^D \sum_{c \in C'} b_{c,j}^C X_{c+1}^{\mathcal{C},M}$, where $i, j \in I^{\mathcal{S}}$, and $i \neq j$. Next,

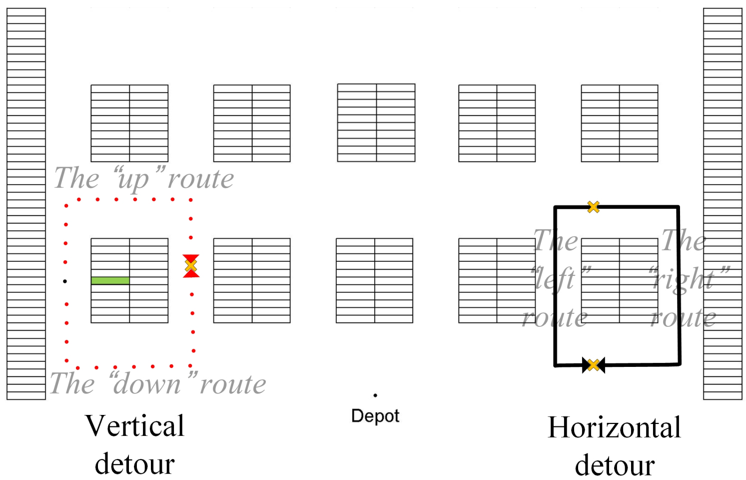


Figure 4.2: Examples of a vertical (horizontal) detour

let $l^{P,U}_{i,j} = a^U_{i,j} \sum_{r \in R \setminus R'} b^R_{r,j} Y^{\mathcal{R},M}_{r-1}$ and $l^{P,D}_{i,j} = a^D_{i,j} \sum_{r \in R \setminus R'} b^R_{r,j} Y^{\mathcal{R},M}_{r+1}$, where $i, j \in I^{\mathcal{S}}$, and $i \neq j$. Then let $l^{S,L}_{i,j} = a^L_{i,j} x^{\mathcal{S}}_j$ and $l^{S,R}_{i,j} = a^R_{i,j} x^{\mathcal{S}}_j$, where $i, j \in I^{\mathcal{S}}$, and $i \neq j$. Finally, $l^{S,U}_{i,j} = a^U_{i,j} y^{\mathcal{S}}_j$ and $l^{S,D}_{i,j} = a^D_{i,j} y^{\mathcal{S}}_j$, where $i, j \in I^{\mathcal{S}}$, and $i \neq j$. Later on details in the formulations will be explained.

Table 4.2 summarizes the decision variable notations of APPT-VRP.

4.3.3 MILP formulation

Due to the length of the MILP formulation for APPT-VRP, we break the formulation into two parts. The first part, the routing behavior formulations, contains the objective function, timing established constraints, coordination constraints, tour constraints, and payload constraints. The second part, the hand-off location formulations, contains the constraints related to item hand-off location determination, the distance determination, and linearization related constraints.

Table 4.2: Summary of decision variable notations

$a_{v_1,v_2,i} \in \{0,1\}$	$a_{v_1,v_2,i} = 1$ if picker $v_1 \in P$ passes item $i \in I$ transporter $v_2 \in D$ at a particular location $j = i + \lvert I\rvert \in I^{\mathcal{S}}$ after the item is picked up at location i by v_1.
$a^L_{i,j}$, $a^R_{i,j} \in \{0,1\})$	$a^L_{i,j} = 1$ ($a^R_{i,j} = 1$) if vehicle travels through the left (right) route on detour from i to j when $b^{\text{detour},X}_{i,j} = 1$, where $i \in N$, $j \in \{N : i \in I^{\mathcal{S}} \mid I^{\mathcal{S}} : i \in N \backslash I^{\mathcal{S}}\}$, and $i \neq j$.
$a^U_{i,j}$, $a^D_{i,j} \in \{0,1\}$	$a^U_{i,j} = 1$ ($a^D_{i,j} = 1$) if vehicles travel through the up (down) route on detour from i to j when $b^{\text{detour},Y}_{i,j} = 1$, where $i \in N$, $j \in \{N : i \in I^{\mathcal{S}} \mid I^{\mathcal{S}} : i \in N \backslash I^{\mathcal{S}}\}$, and $i \neq j$.
$b^{\mathcal{C}}_{c,i} \in \{0,1\}$	$b^{\mathcal{C}}_{c,i} = 1$ if item $i - \lvert I\rvert$ is passed within column c, where $c \in C$ and $i \in I^{\mathcal{S}}$.
$b^{\mathcal{C},\text{Same}}_{i,j} \in \{0,1\}$	$b^{\mathcal{C},\text{Same}}_{i,j}$ if node i and node j are located in the same column $c \in C'$.
$b^{\text{detour},X}_{i,j}$, $b^{\text{detour},Y}_{i,j} \in \{0,1\}$	$b^{\text{detour},X}_{i,j} = 1$ ($b^{\text{detour},Y}_{i,j} = 1$) if the distance between node i and node j is longer than the difference between the x (y) coordinates.
$b^{\mathcal{R}}_{r,i} \in \{0,1\}$	$b^{\mathcal{R}}_{r,i} = 1$ if item $i - \lvert I\rvert$ is passed within row r, where $i \in I^{\mathcal{S}}$ and $r \in R$.
$b^{\mathcal{R},\text{Same}}_{i,j} \in \{0,1\}$	$b^{\mathcal{R},\text{Same}}_{i,j} = 1$ if node i and node j are located in the same row $r \in R \backslash R'$.
$d^X_{i,j}$, $d^Y_{i,j} \geq 0$	The x (y) distance from i to j, where $i \in N$, $j \in \{N : i \in I^{\mathcal{S}} \mid I^{\mathcal{S}} : i \in N \backslash I^{\mathcal{S}}\}$, and $i \neq j$.
$l^{P,L}_{i,j}$, $l^{P,R}_{i,j} \geq 0$	$l^{P,L}_{i,j} = a^U_{i,j} \sum_{c \in C'} b^C_{c,j} X^{\mathcal{C},M}_{c-1}$ and $l^{P,R}_{i,j} = a^D_{i,j} \sum_{c \in C'} b^C_{c,j} X^{\mathcal{C},M}_{c+1}$ to linearize Constraints (4.43) – (4.46), where $i, j \in I^{\mathcal{S}}$, and $i \neq j$.
$l^{P,U}_{i,j}$, $l^{P,D}_{i,j} \geq 0$	$l^{P,U}_{i,j} = a^U_{i,j} \sum_{r \in R \backslash R'} b^R_{r,j} Y^{\mathcal{R},M}_{r-1}$ and $l^{P,D}_{i,j} = a^D_{i,j} \sum_{r \in R \backslash R'} b^R_{r,j} Y^{\mathcal{R},M}_{r+1}$ to linearize Constraints (4.43) – (4.46), where $i, j \in I^{\mathcal{S}}$, and $i \neq j$.
$l^{S,L}_{i,j}$, $l^{S,R}_{i,j} \geq 0$	$l^{S,L}_{i,j} = a^L_{i,j} x^{\mathcal{S}}_j$ and $l^{S,R}_{i,j} = a^R_{i,j} x^{\mathcal{S}}_j$ to linearize Constraints (4.43) – (4.46), where $i, j \in I^{\mathcal{S}}$, and $i \neq j$.
$l^{S,U}_{i,j}$, $l^{S,D}_{i,j} \geq 0$	$l^{S,U}_{i,j} = a^U_{i,j} y^{\mathcal{S}}_j$ and $l^{S,D}_{i,j} = a^D_{i,j} y^{\mathcal{S}}_j$ to linearize Constraints (4.43) – (4.46), where $i, j \in I^{\mathcal{S}}$, and $i \neq j$.
$t_{v,j} \geq 0$	The arrival time at node $j \in \Delta^+_v$ for vehicle $v \in V$.
$w_{v,i,j} \geq 0$	The total weight of items carried by transporter $v \in D$ after leaving node j, having traveled from node i.
$x_{v,i,j}$, $y_{v,i,j} \in \{0,1\}$	$x_{v,i,j} = 1$ ($y_{v,i,j} = 1$) if picker $v \in P$ (transporter $v \in D$) travels from node $i \in \Delta^-_{v,j}$ to node $j \in \Delta^+_v$.
$x^{\mathcal{S}}_i$, $y^{\mathcal{S}}_i \geq 0$	The x (y) coordinate of the item hand-off location for item i, where $i \in I^{\mathcal{S}}$, and $d^X_{i,j}$ ($d^Y_{i,j}$) represents the x (y) distance from i to j, where $i \in N$, $j \in \{N : i \in I^{\mathcal{S}} \mid I^{\mathcal{S}} : i \in N \backslash I^{\mathcal{S}}\}$, and $i \neq j$.

The routing behavior formulations

According to [41], optimizing the warehouse service level in terms of order retrieval times is a need for any order picking systems. The sooner an order can be retrieved, the earlier the order can be shipped to the customer with less chances to miss the shipping due time. Moreover, shorter retrieval times imply better flexibility in handling late order changes. Therefore, the objective of the problem is to minimize the makespan, which is the latest time at which all items from the pick list I are dropped off at the packing station. The makespan is represented by continuous decision variable $m \geq 0$.

The first part of the MILP formulation is as follows.

$$\text{Min} \quad m \tag{4.1}$$

$$\text{s.t.} \quad m \geq t_{v,j} \quad \forall\, v \in D, j \in \Delta_v^*, \tag{4.2}$$

$$\sum_{v \in P} \sum_{i \in \Delta_{v,j}^-} x_{v,i,j} = 1 \quad \forall\, j \in I \cup I^{\mathcal{S}}, \tag{4.3}$$

$$\sum_{v \in D} \sum_{i \in \Delta_{v,j}^-} y_{v,i,j} = 1 \quad \forall\, j \in I^{\mathcal{S}}, \tag{4.4}$$

$$2a_{v_1,v_2,i} \leq \sum_{i \in \Delta_{v_1,i+|I|}^-} x_{v_1,i,i+|I|} + \sum_{k \in \Delta_{v_2,i+|I|}^-} y_{v_2,k,i+|I|} \quad \forall\, i \in I, v_1 \in P, v_2 \in D, \tag{4.5}$$

$$a_{v_1,v_2,i} + 1 \geq \sum_{i \in \Delta_{v_1,i+|I|}^-} x_{v_1,i,i+|I|} + \sum_{k \in \Delta_{v_2,i+|I|}^-} y_{v_2,k,i+|I|} \quad \forall\, i \in I, v_1 \in P, v_2 \in D, \tag{4.6}$$

$$\sum_{v_1 \in P} \sum_{v_2 \in D} a_{v_1,v_2,j} = 1 \quad \forall\, j \in I, \tag{4.7}$$

$$t_{v_2,i+|I|} \geq t_{v_1,i} + s_{v_1,i}^{\text{pick}} + (d_{i,i+|I|}^X + d_{i,i+|I|}^Y)/F_{v_1} - M(1 - a_{v_1,v_2,i})$$
$$\forall\, v_1 \in P, v_2 \in D, i \in I, \tag{4.8}$$

$$t_{v_1,k} \geq t_{v_2,j=i+|I|} + s_{v_1,i+|I|}^{\text{place}} + (d_{i+|I|,k}^X + d_{i+|I|,k}^Y)/F_{v_1} - M(2 - a_{v_1,v_2,i} - x_{v_1,i+|I|,k})$$
$$\forall\, v_1 \in P, v_2 \in D, i \in I, k \in \left\{\Delta_{v_1}^+ : \{i + |I|\} \in \Delta_{v_1,i+|I|}^-\right\}, \tag{4.9}$$

$$\sum_{j \in \{\Delta_v^+ : \Delta_v^0 \in \Delta_{v,j}^-\}} x_{v,\Delta_v^0,j} \leq 1 \quad \forall\, v \in P, \tag{4.10}$$

$$\sum_{i \in \Delta_{v,j}^-} x_{v,i,j} = x_{v,j,j+|I|} \quad \forall\, v \in P, j \in I, \tag{4.11}$$

$$|I| \sum_{j \in \{\Delta_v^+ : \Delta_v^0 \in \Delta_{v,j}^-\}} x_{v,\Delta_v^0,j} \geq \sum_{k \in \{\Delta_v^+ : j \in \Delta_{v,k}^-\}} x_{v,j,k} \quad \forall\, v \in P, j \in I, \tag{4.12}$$

$$x_{v,i,i+|I|} \geq \sum_{k \in \{\Delta_v^+ : i+|I| \in \Delta_{v,k}^-\}} x_{v,i+|I|,k} \quad \forall\, v \in P, i \in I, \tag{4.13}$$

$$\sum_{i \in \Delta_{v,j}^-} x_{v,i,j} \leq 1 \quad \forall\, v \in P, j \in \Delta_v^+, \tag{4.14}$$

$$\sum_{j \in \{\Delta_v^+ : i \in \Delta_{v,j}^-\}} x_{v,i,j} \leq 1 \quad \forall\, v \in P, i \in \{\Delta_v^+ \cup \Delta_v^0\}, \tag{4.15}$$

$$\sum_{j \in \{\Delta_v^+ : \Delta_v^0 \in \Delta_{v,j}^-\}} y_{v,\Delta_v^0,j} = 1 \quad \forall\, v \in D, \tag{4.16}$$

$$\sum_{i \in \Delta_{v,j}^-} y_{v,i,j} = 1 \quad \forall\, v \in D, j \in \Delta_v^*, \tag{4.17}$$

$$\sum_{i \in \Delta_{v,j}^-} y_{v,i,j} = \sum_{k \in \{\Delta_v^+ : j \in \Delta_{v,k}^-\}} y_{v,j,k} \quad \forall\, v \in D, j \in \Delta_v^+, \tag{4.18}$$

$$\sum_{i \in \Delta_{v,j}^-} y_{v,i,j} \leq 1 \quad \forall\, v \in D, j \in \Delta_v^+, \tag{4.19}$$

$$\sum_{j \in \{\Delta_v^+ : i \in \Delta_{v,j}^-\}} y_{v,i,j} \leq 1 \quad \forall\, v \in D, i \in \{\Delta_v^+ \cup \Delta_v^0\}, \tag{4.20}$$

$$t_{v,0} = 0 \quad \forall\, v \in V, \tag{4.21}$$

$$t_{v,i} \geq t_{v,\Delta_v^0} + (d_{\Delta_v^0,i}^X + d_{\Delta_v^0,i}^Y)/F_v - M(1 - x_{v,\Delta_v^0,i}) \quad \forall\, v \in P, i \in I, \tag{4.22}$$

$$t_{v,j} \geq t_{v,i} + s_{v,i}^{\text{pick}} + (d_{i,j}^X + d_{i,j}^Y)/F_{v_1} - M(1 - x_{v,i,j})$$
$$\forall\, v \in P, j \in \Delta_v^+, i \in \{\Delta_{v,j}^- \cap I\}, \tag{4.23}$$

$$t_{v,k} \geq t_{v,j=i+|I|} + s_{v,i+|I|}^{\text{place}} + (d_{i+|I|,k}^X + d_{i+|I|,k}^Y)/F_v - M(1 - x_{v,i+|I|,k})$$
$$\forall\, v \in P, i \in I, k \in \left\{\Delta_{v_1}^+ : \{i + |I|\} \in \Delta_{v_1,i+|I|}^-\right\}, \tag{4.24}$$

$$t_{v,k} \geq t_{v,i+|I|} + \sum_{v' \in P} s_{v',i}^{\text{place}} a_{v',v,i} + (d_{i+|I|,k}^X + d_{i+|I|,k}^Y)/F_v - M(1 - y_{v,j,k})$$
$$\forall\, v \in D, k \in \Delta_v^+ \backslash \{i + |I|\}, i \in I, \tag{4.25}$$

$$t_{v,j} \geq t_{v,i} + (d_{i,j}^X + d_{i,j}^Y)/F_v - M(1 - y_{v,i,j})$$
$$\forall\, v \in D, j \in \Delta_v^+, i \in \{\Delta_v^* \cup 0 : i \in \Delta_{v,j}^-\}, \tag{4.26}$$

$$t_{v,j_2} \geq t_{v,j_1} + s_v^{\text{drop}} \left(\sum_{i \in \{\Delta^-_{v,j_1} \setminus (\Delta^*_v \cap j_2)\}} y_{v,i,j_1} \right) + (d^X_{j_1,j_2} + d^Y_{j_1,j_2})/F_v - M(1 - y_{v,j_1,j_2})$$

$$\forall\ v \in D, j_1 \in \{\Delta^*_v \setminus \max\{\Delta^*_v\}\}, j_2 \in \{\Delta^+_v \setminus j_1\}, \tag{4.27}$$

$$w_{v,i,j} \leq \hat{w}_v y_{v,i,j} \quad \forall\ v \in D, j \in \Delta^+_v, i \in \Delta^-_{v,j}, \tag{4.28}$$

$$w_{v,\Delta^0_v,i+|I|} = (w'_v + \bar{w}_{i+|I|}) y_{v,\Delta^0_v,i+|I|} \quad \forall\ v \in D, j \in \{I^{\mathcal{S}} \cap \Delta^+_v\}, \tag{4.29}$$

$$w_{v,i,j} = \bar{w}_j y_{v,i,j} \quad \forall\ v \in D, j \in I^{\mathcal{S}}, i \in \{\Delta^-_{v,j} \cap \Delta^*_v\}, \tag{4.30}$$

$$w_{v,j,k} \geq \sum_{\substack{i \in \Delta^-_{v,j} \\ i \neq k}} w_{v,i,j} + \bar{w}_k y_{v,j,k} - \hat{w}_v (1 - y_{v,j,k}) \quad \forall\ v \in D, k \in I^{\mathcal{S}}, j \in \{I^{\mathcal{S}} : k \neq j\}, \tag{4.31}$$

The objective function (4.1) seeks to minimize the makespan. Constraint (4.2) serves to establish the makespan. Constraints (4.3) and (4.4) ensure that each item is retrieved by a picker and a transporter. This requires each item storage location and each item hand-off location to be visited by a picker, and each item hand-off location to be visited by a transporter. Constraints (4.5)–(4.9) coordinate the picker and transporter at location j, where Constraints (4.5), (4.6), and (4.7) ensure $a_{v_1,v_2,i} = 1$ to pair picker v_1 with transporter v_2 to retrieve item i, while Constraints (4.8) and (4.9) establish the timing of this coordination. In particular, Constraint (4.5) forces $a_{v_1,v_2,i} = 0$ unless picker v_1 and transporter v_2 meet at the location $i + |I|$ to pass an item after v_1 picked up the item at i. Conversely, Constraint (4.6) sets $a_{v_1,v_2,i} = 1$ if v_1 picks item i and then meets v_2 at $i + |I|$. Constraint (4.7) ensures that each item is associated with exactly one picker/transporter pair. Next, Constraint (4.8) specifies that a transporter may retrieve an item after a picker has completed the picking operation and moved to the hand-off location. Constraint (4.9) prohibits a picker from moving to the next location, k, until the placement of an item at $i + |I|$ is completed. The value of M represents a sufficiently large number (an upper bound on the makespan).

Valid vehicle routes are established by Constraints (4.10)–(4.15). Constraint (4.10) ensures at most one picker to start its tour by departing from its initial location. Constraint

(4.11) conserves the flow of pickers. Constraint (4.12) ensures a picker must visit the hand-off location if it leaves the picking location. Constraint (4.13) ensures a picker must go out of the depot if it is used. Constraints (4.14) and (4.15) ensure picker from visiting or leaving any node no more than once, respectively. Constraints (4.16)–(4.20) are analogous to Constraints (4.10)–(4.15) for the case of transporters.

Constraints (4.21)–(4.24) establish travel time into the routing process, where Constraint (4.22) initializes all vehicles starting at time zero. Constraint (4.21) establishes the arrival time for pickers from the depot to a picking location. Constraint (4.23) ensures if a picker travels from $i \in I$ to $j = i + |I|$, then the arrival time to j cannot be before the arrival time to i plus the pick up time at i plus the travel time from i to j. Constraint (4.24) establishes the arrival time for pickers from a hand-off location to a picking location. Similarly for transporters, Constraint (4.25) guarantees that transporter's arrival time to k cannot be earlier than the summation of the arrival time to j, the placement service time performed by the partnering picker at j, and the travel time from j to k. Constraint (4.26) ensures valid start times when a transporter leaves a depot replica, while Constraint (4.27) establishes the drop-off time required before visiting subsequent locations.

Payload limitations for transporters are addressed by Constraints (4.28)–(4.31). As in Lee and Murray [100], $w_{v,j,k} = 0$ unless v travels directly from j to k, and pickers do not have explicit payload capacity constraints, since they may only retrieve one item at a time; if item $i \in I$ is too heavy for picker $v \in P$, then $i \notin \Delta_v^+$ and $i \notin \Delta_{v,j}^-$. Constraint (4.28) ensures that the transporter $v \in D$ not going to be overloaded after visiting node j. Constraint (4.29) addresses the situation that when transporter leaves its initial location and travels to some location j, the total weight carried by transporter equals the summation of the initial weight w'_v and the quantity picked up at location j. Constraint (4.30) makes sure the payload weight carried by transporter v when picking up the first item after unloaded at a depot replica. Constraint (4.31) forces the payload weight to be not less than the summation of item weight when $v \in D$ leaves j, and the weight loaded at k (when v leaves k from j).

The hand-off location formulations

The second part of the MILP formulation is as follows.

$$\sum_{c \in C} b^{\mathcal{C}}_{c,i} = 1 \quad \forall\, i \in I^{\mathcal{S}}, \tag{4.32}$$

$$\sum_{r \in R} b^{\mathcal{R}}_{r,i} = 1 \quad \forall\, i \in I^{\mathcal{S}}, \tag{4.33}$$

$$\sum_{r \in R'} b^{\mathcal{R}}_{r,i} \geq \sum_{c \in C'} b^{\mathcal{C}}_{c,i} \quad \forall\, i \in I^{\mathcal{S}}, \tag{4.34}$$

$$x^{\mathcal{S}}_i - \sum_{c \in C'} 0.5 W^C_c b^{\mathcal{C}}_{c,i} \leq \sum_{c \in C} b^{\mathcal{C}}_{c,i} X^{\mathcal{C},M}_c \leq x^{\mathcal{S}}_i + \sum_{c \in C'} 0.5 W^C_c b^{\mathcal{C}}_{c,i} \quad \forall\, i \in I^{\mathcal{S}}, \tag{4.35}$$

$$y^{\mathcal{S}}_i - \sum_{r \in \{R \setminus R'\}} 0.5 H^R_r b^{\mathcal{R}}_{r,i} \leq \sum_{r \in R} b^{\mathcal{R}}_{r,i} Y^{\mathcal{R},M}_r \leq y^{\mathcal{S}}_i + \sum_{r \in \{R \setminus R'\}} 0.5 H^R_r b^{\mathcal{R}}_{r,i} \quad \forall\, i \in I^{\mathcal{S}}, \tag{4.36}$$

$$a^L_{i,j} + a^R_{i,j} = b^{\text{detour},X}_{i,j} \quad \forall\, i \in N, j \in \{N \setminus I^{\mathcal{S}} : i \in I^{\mathcal{S}} | I^{\mathcal{S}} : i \in N \setminus I^{\mathcal{S}}\}, i \neq j, \tag{4.37}$$

$$a^U_{i,j} + a^D_{i,j} = b^{\text{detour},Y}_{i,j} \quad \forall\, i \in N, j \in \{N \setminus I^{\mathcal{S}} : i \in I^{\mathcal{S}} | I^{\mathcal{S}} : i \in N \setminus I^{\mathcal{S}}\}, i \neq j, \tag{4.38}$$

$$d^X_{i,j} \geq x^{\mathcal{S}}_i - X^{\mathcal{K}}_j + 2a^L_{i,j}\Big(X^{\mathcal{K}}_j - \sum_{c \in C'} B^C_{c,j} X^{\mathcal{C},M}_{c-1}\Big) \quad \forall\, i \in N, j \in \{N \setminus I^{\mathcal{S}} : i \in I^{\mathcal{S}} | I^{\mathcal{S}} : i \in N \setminus I^{\mathcal{S}}\}, \tag{4.39}$$

$$d^X_{i,j} \geq X^{\mathcal{K}}_j - x^{\mathcal{S}}_i + 2a^R_{i,j}\Big(\sum_{c \in C'} B^C_{c,j} X^{\mathcal{C},M}_{c+1} - X^{\mathcal{K}}_j\Big) \quad \forall\, i \in N, j \in \{N \setminus I^{\mathcal{S}} : i \in I^{\mathcal{S}} | I^{\mathcal{S}} : i \in N \setminus I^{\mathcal{S}}\}, \tag{4.40}$$

$$d^Y_{i,j} \geq y^{\mathcal{S}}_i - Y^{\mathcal{K}}_j + 2a^D_{i,j}\Big(Y^{\mathcal{K}}_j - \sum_{r \in R \setminus R'} B^R_{r,j} Y^{\mathcal{R},M}_{r+1}\Big) \quad \forall\, i \in N, j \in \{N \setminus I^{\mathcal{S}} : i \in I^{\mathcal{S}} | I^{\mathcal{S}} : i \in N \setminus I^{\mathcal{S}}\}, \tag{4.41}$$

$$d^Y_{i,j} \geq Y^{\mathcal{K}}_j - y^{\mathcal{S}}_i + 2a^U_{i,j}\Big(\sum_{r \in R \setminus R'} B^R_{r,j} Y^{\mathcal{R},M}_{r-1} - Y^{\mathcal{K}}_j\Big) \quad \forall\, i \in N, j \in \{N \setminus I^{\mathcal{S}} : i \in I^{\mathcal{S}} | I^{\mathcal{S}} : i \in N \setminus I^{\mathcal{S}}\}, \tag{4.42}$$

$$d^X_{i,j} \geq x^{\mathcal{S}}_i - x^{\mathcal{S}}_j + 2a^L_{i,j}\Big(x^{\mathcal{S}}_j - \sum_{c \in C'} b^C_{c,j} X^{\mathcal{C},M}_{c+1}\Big) \quad \forall\, i, j \in I^{\mathcal{S}}, i \neq j, \tag{4.43}$$

$$d^X_{i,j} \geq x^{\mathcal{S}}_j - x^{\mathcal{S}}_i + 2a^R_{i,j}(\sum_{c\in C'} b^C_{c,j} X^{\mathcal{C},M}_{c-1} - x^{\mathcal{S}}_j) \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.44}$$

$$d^Y_{i,j} \geq y^{\mathcal{S}}_i - y^{\mathcal{S}}_j + 2a^D_{i,j}(y^{\mathcal{S}}_j - \sum_{r\in R\setminus R'} b^R_{r,j} Y^{\mathcal{R},M}_{r+1}) \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.45}$$

$$d^Y_{i,j} \geq y^{\mathcal{S}}_j - y^{\mathcal{S}}_i + 2a^U_{i,j}(\sum_{r\in R\setminus R'} b^R_{r,j} Y^{\mathcal{R},M}_{r-1} - y^{\mathcal{S}}_j) \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.46}$$

$$2b^{\mathcal{C},\text{Same}}_{i,j} \leq b^C_{c,i} + B^C_{c,j} \leq 1 + b^{\mathcal{C},\text{Same}}_{i,j} \quad \forall\, c \in C', i \in N, j \in \{N\setminus I^{\mathcal{S}} : i \in I^{\mathcal{S}} | I^{\mathcal{S}} : i \in N\setminus I^{\mathcal{S}}\}, \tag{4.47}$$

$$2b^{\mathcal{R},\text{Same}}_{i,j} \leq b^R_{r,i} + B^R_{r,j} \leq 1 + b^{\mathcal{R},\text{Same}}_{i,j} \quad \forall\, r \in R\setminus R', i \in N, j \in \{N\setminus I^{\mathcal{S}} : i \in I^{\mathcal{S}} | I^{\mathcal{S}} : i \in N\setminus I^{\mathcal{S}}\}, \tag{4.48}$$

$$2b^{\mathcal{C},\text{Same}}_{i,j} \leq b^C_{c,i} + b^C_{c,j} \leq 1 + b^{\mathcal{C},\text{Same}}_{i,j} \quad \forall\, c \in C', i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.49}$$

$$2b^{\mathcal{R},\text{Same}}_{i,j} \leq b^R_{r,i} + b^R_{r,j} \leq 1 + b^{\mathcal{R},\text{Same}}_{i,j} \quad \forall\, r \in R\setminus R', i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.50}$$

$$2b^{\text{detour},X}_{i,j} \leq (1 - b^{\mathcal{R},\text{Same}}_{i,j}) + b^{\mathcal{C},\text{Same}}_{i,j} \leq 1 + b^{\text{detour},X}_{i,j} \quad \forall\, i \in N, j \in \{N\setminus I^{\mathcal{S}} : i \in I^{\mathcal{S}} | I^{\mathcal{S}} : i \in N\setminus I^{\mathcal{S}}\}, \tag{4.51}$$

$$2b^{\text{detour},Y}_{i,j} \leq b^{\mathcal{R},\text{Same}}_{i,j} + (1 - b^{\mathcal{C},\text{Same}}_{i,j}) \leq 1 + b^{\text{detour},Y}_{i,j} \quad \forall\, i \in N, j \in \{N\setminus I^{\mathcal{S}} : i \in I^{\mathcal{S}} | I^{\mathcal{S}} : i \in N\setminus I^{\mathcal{S}}\}, \tag{4.52}$$

$$2b^{\text{detour},X}_{i,j} \leq (1 - b^{\mathcal{R},\text{Same}}_{i,j}) + b^{\mathcal{C},\text{Same}}_{i,j} \leq 1 + b^{\text{detour},X}_{i,j} \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.53}$$

$$2b^{\text{detour},Y}_{i,j} \leq b^{\mathcal{R},\text{Same}}_{i,j} + (1 - b^{\mathcal{C},\text{Same}}_{i,j}) \leq 1 + b^{\text{detour},Y}_{i,j} \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.54}$$

Constraints (4.32)–(4.35) ensure item hand-off locations can only happen in aisles. Constraint (4.32) ensures all hand-off locations located in a column and a row. Constraints (4.33) and (4.34) make sure that item hand-off is only allowed on aisles. Constraints (4.35) ensures the x coordinate of the hand-off location of item i equal to the x coordinate of the middle point in column c if the hand-off location is in the column. Constraints (4.36) is analogous to Constraints (4.35) for the case of rows.

Detour constraints are introduced in Constraints (4.37)–(4.41). Constraint (4.37) makes sure only traveling from location i to location j either along the left or the right route on detour when $b_{i,j}^{\text{detour},X} = 1$. Constraint (4.38) is analogous to Constraint (4.37) for traveling along either the up or the down route when $b_{i,j}^{\text{detour},Y} = 1$. Constraints (4.39) determines the absolute horizontal distance between the hand-off location of item i and known location j. Note that W and H are functioning as big-M values in the model. Constraint (4.41) determines the vertical traveling distance between the hand-off location of item i and the known location j.

Constraints (4.43) – (4.46) determine the shortest distance between two locations, where only one of these two is an item hand-off location. Constraints (4.43) – (4.46) are analogous to Constraints (4.39) – (4.41) for determining the traveling distances between two item hand-off locations. Constraints (4.43) and (4.44) determine the absolute horizontal distance between two item hand-off locations. Constraints (4.45) and (4.46) determine the vertical traveling distance between the hand-off locations of items i and j.

Constraints (4.47) – (4.52) determine if any two locations are on the same row/column. Constraints (4.47) – (4.50) determine if any two locations are located in the same row/column in order to determine a detour is required for vehicles to travel from one to another. Then Constraints (4.51) – (4.54) determine if the distance between nodes i and j is longer than horizontal or vertical Manhattan distance. In Constraint (4.51), $b_{i,j}^{\text{detour},X} = 1$ if a known location i and item hand-off location j are located in the same row but different columns. In Constraint (4.52), $b_{i,j}^{\text{detour},Y} = 1$ if a known location i and item hand-off location j are located in the same even column $c \in C'$ but different rows. Constraints (4.53) – (4.54) are analogous to Constraints (4.51) – (4.52) for determining if the distance between two item hand-off locations is longer than horizontal or vertical Manhattan distance.

Note that Constraints (4.43) – (4.46) are nonlinear because the coordinates of item hand-off locations are decision variables. Thus, by introducing $l_{i,j}^{S,L}, l_{i,j}^{S,R}, l_{i,j}^{P,L}, l_{i,j}^{P,R}, l_{i,j}^{S,U}, l_{i,j}^{S,D}, l_{i,j}^{P,U}$, and $l_{i,j}^{P,D}$, Constraints (4.43) – (4.46) could be linearized as Constraints (4.55) – (4.74) shown below. Constraints (4.55) – (4.63) determine the values of the new introduced variables, and then Constraints (4.43) – (4.46) could be reformulated as Constraints (4.71) – (4.74).

$$l_{i,j}^{S,L} \leq W a_{i,j}^{L} \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.55}$$

$$x_j^{\mathcal{S}} - W(1 - a_{i,j}^{L}) \leq l_{i,j}^{S,L} \leq x_j^{\mathcal{S}} \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.56}$$

$$l_{i,j}^{S,R} \leq W a_{i,j}^{R} \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.57}$$

$$x_j^{\mathcal{S}} - W(1 - a_{i,j}^{R}) \leq l_{i,j}^{S,R} \leq x_j^{\mathcal{S}} \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.58}$$

$$l_{i,j}^{S,U} \leq H a_{i,j}^{U} \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.59}$$

$$y_j^{\mathcal{S}} - H(1 - a_{i,j}^{U}) \leq l_{i,j}^{S,U} \leq y_j^{\mathcal{S}} \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.60}$$

$$l_{i,j}^{S,D} \leq H a_{i,j}^{D} \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.61}$$

$$y_j^{\mathcal{S}} - H(1 - a_{i,j}^{D}) \leq l_{i,j}^{S,D} \leq y_j^{\mathcal{S}} \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.62}$$

$$l_{i,j}^{P,L} \leq W a_{i,j}^{L} \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.63}$$

$$\sum_{c \in C'} b_{c,j}^{C} X_{c+1}^{\mathcal{C},M} - W(1 - a_{i,j}^{L}) \leq l_{i,j}^{P,L} \leq \sum_{c \in C'} b_{c,j}^{C} X_{c+1}^{\mathcal{C},M} \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.64}$$

$$l_{i,j}^{P,R} \leq W a_{i,j}^{R} \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.65}$$

$$\sum_{c \in C'} b_{c,j}^{C} X_{c-1}^{\mathcal{C},M} - W(1 - a_{i,j}^{R}) \leq l_{i,j}^{P,R} \leq \sum_{c \in C'} b_{c,j}^{C} X_{c-1}^{\mathcal{C},M} \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.66}$$

$$l_{i,j}^{P,U} \leq H a_{i,j}^{U} \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.67}$$

$$\sum_{r \in R \setminus R'} b_{r,j}^{R} Y_{r-1}^{\mathcal{R},M} - H(1 - a_{i,j}^{U}) \leq l_{i,j}^{P,U} \leq \sum_{r \in R \setminus R'} b_{r,j}^{R} Y_{r-1}^{\mathcal{R},M} \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.68}$$

$$l_{i,j}^{P,D} \leq H a_{i,j}^{D} \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.69}$$

$$\sum_{r \in R \setminus R'} b_{r,j}^{R} Y_{r+1}^{\mathcal{R},M} - H(1 - a_{i,j}^{D}) \leq l_{i,j}^{P,D} \leq \sum_{r \in R \setminus R'} b_{r,j}^{R} Y_{r+1}^{\mathcal{R},M} \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.70}$$

$$d_{i,j}^{X} \geq x_i^{\mathcal{S}} - x_j^{\mathcal{S}} + 2(l_{i,j}^{S,L} - l_{i,j}^{P,L}) \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.71}$$

$$d_{i,j}^{X} \geq x_j^{\mathcal{S}} - x_i^{\mathcal{S}} + 2(l_{i,j}^{P,R} - l_{i,j}^{S,R}) \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.72}$$

$$d_{i,j}^{Y} \geq y_i^{\mathcal{S}} - y_j^{\mathcal{S}} + 2(l_{i,j}^{S,D} - l_{i,j}^{P,D}) \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.73}$$

$$d_{i,j}^{Y} \geq y_j^{\mathcal{S}} - y_i^{\mathcal{S}} + 2(l_{i,j}^{P,U} - l_{i,j}^{S,U}) \quad \forall\, i,j \in I^{\mathcal{S}}, i \neq j, \tag{4.74}$$

4.4 The proposed APPT-VRP heuristic

As discussed in Lee and Murray [100], efficient heuristics for larger-scale problems are desirable. Moreover, in our preliminary test, the APPT-VRP with 10 items took longer than 24 hours to solve the problem optimally using mixed integer linear programming (MILP) solvers owing to the determination of the transshipment nodes. Thus, an iterative heuristic is proposed, and the pseudo-code of the main function is provided in Algorithm 5. The proposed heuristic begins with the construction phase (lines 2–5 in Algorithm 5), which provides initial routing solutions, and then local search procedure (lines 6–22 in Algorithm 5) is applied to improve the quality of the solution by exploring neighboring solutions. In the initialization phase, first, the hand-off locations are set as item storage locations ($x_i^{\mathcal{S}} \leftarrow X_i^{\mathcal{K}}$ and $y_i^{\mathcal{S}} \leftarrow Y_i^{\mathcal{K}}$), then the visiting order of locations ($x_{v,i,j}$) for each picker is determined, and finally the corresponding transporters' visiting orders ($y_{v,i,j}$) are determined based on the found pickers' orders. Note that once these variables are determined, every variables in the MILP model could be ascertained except $t_{v,j}$. Thus, after the initial routing plan is determined, a linear programming (LP) relaxation of the APPT-VRP model, which we only solve using the linear variables of arrival times ($t_{v,j}$) by feeding all other ascertained variables into the APPT-VRP model, is applied to determine the visiting time at every location for every vehicle before the improvement phase.

In the improvement phase, first, we apply function *RouteLocalSearch()*, incorporating insertion and exchange neighborhood search procedures to explore if there is any improvement for the neighborhood visiting orders. We search the neighborhood order for pickers first, and then explore the best corresponding neighborhood order of transporters. Once the new neighborhood visiting order is conducted, the new visiting time is determined by the LP relaxation model, and a search procedure is applied to update the corresponding item hand-off locations based on the new visiting order. If there is no improvement by the proposed neighborhood search procedure, the proposed iterative heuristic re-initializes the

Algorithm 5 Pseudocode for the main APPT-VRP heuristic.

```
Initialize:
HandOffLocation = ItemStorageLocation
[PickerRoute]=NearestNeighbor(HandOffLocation)
[TransporterRoute]=NearestNeighbor(PickerRoute, HandOffLocation)
[ArrivalTime]=LPRelaxation(HandOffLocation, PickerRoute, TransporterRoute)
repeat
    MaxSaving=0
    [ArrivalTime', HandOffLocation', PickerRoute', TransporterRoute']=
      RouteLocalSearch(ArrivalTime, HandOffLocation, PickerRoute,
      TransporterRoute)
      % Algorithm 6
    if max(ArrivalTime)-max(ArrivalTime')=0 then
        Re-initialize ArrivalTime, HandOffLocation, PickerRoute, and TransporterRoute.
    else
        if max(BestArrivalTime)-max(ArrivalTime')≥0 then
            BestArrivalTime=ArrivalTime'; BestHandOffLocation=HandOffLocation';
            BestPickerRoute=PickerRoute'; BestTransporterRoute=TransporterRoute';
        end if
        ArrivalTime=ArrivalTime'; HandOffLocation=HandOffLocation';
        PickerRoute=PickerRoute'; TransporterRoute=TransporterRoute';
    end if
until (Stop)
return BestArrivalTime, BestHandOffLocation, BestPickerRoute,
        BestTransporterRoute
```

visiting order and repeats the neighborhood search to ascertain if better solutions could be found until the stopping criteria are met.

Details of the construction method is presented in Sections 4.4.1. Then the local search procedure is introduced in Section 4.4.2. Within the local search procedure, the search heuristic to optimize the item hand-off locations for vehicles is described in Section 4.4.2.

4.4.1 Construction method for initial solutions

The initial solutions are obtained as follows: first, we initialize the item hand-off locations as the item storage locations. Later on the hand-off locations will be updated to optimize the makespan. Next, the nearest neighbor heuristic, whereby a vehicle route is constructed by starting at the depot and choosing the item storage in the pick list closest to the current location, is applied. Then, based on the solution found for the pickers' routes, transporters' routes are determined. Note that the transporters' routes would be infeasible if a transporter visits location j before location i but a picker visits i before j. In order to generate a feasible solution for the corresponding transporters' routes, the list of available unvisited nodes changes via the nearest neighbor heuristic process.

In the beginning of the nearest neighbor heuristic process for transporters, the list contains only the first visited location of each picker. Then, more locations are released into the list only if the previous location in the visiting order of each picker's route is visited by a transporter. In order to balance the workload for each vehicle in the initial solution, each vehicle can only visit $\lceil |I|/|P| \rceil$ storage locations for pickers and $\lceil |I|/|D| \rceil$ for transporters. If the remaining payload capacity for the transporter is insufficient to accommodate the item at the next hand-off location, the transporter will go to the depot (assigning a depot replica) to unload the batch before visiting the next hand-off location in the route. Once a valid feasible solution of visiting orders for vehicles is conducted, the LP relaxation is applied to determine the arrival time of each location for each vehicle. The LP relaxation only solves $t_{v,j}$ and m in the proposed MILP because by feeding other variables, which can be ascertained based on the visiting orders and the current item hand-off locations, as constants into the MILP.

4.4.2 Improvement phase

In the improvement phase, insertion (lines 3–21 in Algorithm 6) and exchange (lines 22–33 in Algorithm 6) procedures are applied to explore neighborhood routes. Insertion removes one visiting location from the current visiting order of all pickers/transporters, and then the visiting location is inserted into another place in the visiting order. The exchange procedure explores all pairwise exchanges of two visiting locations in the visiting order. Both procedures can be used within either the visiting order of one vehicle or all vehicles of the same type (pickers or transporters).

Explore neighborhood routes

The heuristic searches the neighborhood solutions of visiting order for pickers by insertion and exchange procedures in accordance with Algorithm 6, and then searches the corresponding order for transporters based on the neighborhood solutions for pickers in accordance with function *TransporterRouteSearch()* described in Algorithm 7. As in the construction method, when the number of pickers is equivalent to the number of transporters, the transporters' routes are determined by assigning each transporter to follow a specific picker, in a one-to-one correspondence. Once a feasible neighborhood solution of visiting orders is established, the arrival times are solved via the mentioned LP relaxation. Next, the hand-off locations are updated for the neighborhood solution by function *UpdateHandOffLocations()* described in Algorithm 8. If there is an improvement made by the neighborhood solution, the heuristic explores the new neighborhood solution of the improved neighborhood solution.

Algorithm 6 Pseudocode for the RoutLocalSearch().

```
Require: ArrivalTime, HandOffLocation, PickerRoute, TransporterRoute
ArrivalTime*=ArrivalTime; PickerRoute*=PickerRoute;
TransporterRoute*=TransporterRoute
for All task i ∈ I in PickerRoute do        % Insertion procedure for pickers' routes
    TargetPicker=The Picker which retrieves item i
    for v ∈ P do
        for j= 1: 1 + # of items picked by v do
            if (i is picked by v) and (i!=j or j − 1) then
                PickerRoute'= Move task i prior to j (retrieved by v) in PickerRoute
                [ArrivalTime', HandOffLocation', TransporterRoute']=
                  TransporterRouteSearch(HandOffLocation,PickerRoute',
                  TransporterRoute)   % Algorithm 7
                if max(ArrivalTime')<max(ArrivalTime*)      % Better Mkspn then
                    ArrivalTime*=ArrivalTime'; HandOffLocation=HandOffLocation';
                    PickerRoute*=PickerRoute'; TransporterRoute*=TransporterRoute';
                end if
            end if
        end for
    end for
end for
ArrivalTime=ArrivalTime*; PickerRoute=PickerRoute*;
TransporterRoute=TransporterRoute*
for All task i ∈ I in PickerRoute do        % Exchange procedure for pickers' routes
    for All task j ∈ I, j ≠ i in PickerRoute do
        PickerRoute'=Swap i and j in PickerRoute
        [ArrivalTime', HandOffLocation', TransporterRoute']=
          TransporterRouteSearch(HandOffLocation, PickerRoute',
          TransporterRoute)   % Algorithm 7
        if max(ArrivalTime')<max(ArrivalTime*)      % Better Mkspn then
            ArrivalTime*=ArrivalTime'; HandOffLocation=HandOffLocation';
            PickerRoute*=PickerRoute'; TransporterRoute*=TransporterRoute';
        end if
    end for
end for
return ArrivalTime*, PickerRoute*, TransporterRoute*, HandOffLocation*
```

Algorithm 7 Pseudocode for the TransporterRouteSearch() function.

```
Require: HandOffLocation, PickerRoute, TransporterRoute
if # of pickers = # of transporters then
    HandOffLocation*= ItemStorageLocation
    TransporterRoute* = a transporter follows a picker respectively
    [ArrivalTime*]=LPRelaxation(HandOffLocation*, PickerRoute, TransporterRoute*)
else
    HandOffLocation*=HandOffLocation; TransporterRoute*=TransporterRoute;
    for All task $i \in I^{\mathcal{S}}$ in TransporterRoute do  % Insertion procedure for transporters
        TargetTransporter=The transporter which retrieves item $i$
        for $v \in D$ do
            for $j$= 1: 1 + # of items retrieved by $v$ do
                if ($i$ is retrieved by $v$) and ($i$!=$j$ or $j - 1$) then
                    TransporterRoute'= Move $i$ prior to $j$ (retrieved by $v$)
                        in TransporterRoute
                    if TransporterRoute' is feasible to PickerRoute then
                        [ArrivalTime', HandOffLocation']=
                            UpdateHandOffLocations(HandOffLocation, PickerRoute,
                            TransporterRoute')   % Algorithm 8
                        if max(ArrivalTime')<max(ArrivalTime*) then
                            ArrivalTime*=ArrivalTime';
                            HandOffLocation*=HandOffLocation';
                            TransporterRoute*=TransporterRoute';
                        end if
                    end if
                end if
            end for
        end for
    end for
    ArrivalTime=ArrivalTime*; TransporterRoute=TransporterRoute*
    for All task $i \in I^{\mathcal{S}}$ in TransporterRoute do  % Exchange procedure for transporters
        for All task $j \in I^{\mathcal{S}}$, $j \neq i$ in TransporterRoute do
            TransporterRoute'=Swap $i$ and $j$ in TransporterRoute
            if TransporterRoute' is feasible to PickerRoute then
                [ArrivalTime', HandOffLocation']=
                    UpdateHandOffLocations(PickerRoute, TransporterRoute',
                    HandOffLocation)   % Algorithm 8
                if max(ArrivalTime')<max(ArrivalTime*) then
                    ArrivalTime*=ArrivalTime'; HandOffLocation=HandOffLocation';
                    TransporterRoute*=TransporterRoute';
                end if
            end if
        end for
    end for
end if
return ArrivalTime*, HandOffLocation*, TransporterRoute*
```

Update item hand-off locations

Once a valid neighborhood solution of visiting orders with arrival times is established, the procedure described by Algorithm 8 is proposed to update the hand-off locations one at a time chronologically based on arrival times solved by the mentioned LP relaxation. Recall that unlike our previous work, this approach allows pickers to move toward a transporter to pass the item in order to reduce the traveling routes for transporters.

For this purpose, we need to determine a better item hand-off location i if picker v_1 arrives at i earlier than transporter v_2 when $t_{v_1,i} < t_{v_2,i}$, where $t_{v,i}$ represents the arrival time of location i for vehicle v. In such a case, the transporter has to travel $(t_{v_2,i} - t_{v_1,i})F_{v_2}$ while the picker is waiting, where F_v represents the speed of vehicle v. Let AdditionalTravelDistanceForv_1 (ReducedTravelDistanceForv_2) represent the additional (reduced) travel distance for v_1 (v_2) to meet without either vehicle to wait, respectively. Thus,

$$\text{AdditionalTravelDistanceFor}v_1 + \text{ReducedTravelDistanceFor}v_2 = (t_{v_2,i} - t_{v_1,i})F_{v_2},$$

and we know that

$$(\text{AdditionalTravelDistanceFor}v_1)F_{v_1} = (\text{ReducedTravelDistanceFor}v_2)F_{v_2}.$$

Therefore,

$$\text{AdditionalTravelDistanceFor}v_1 = (t_{v_2,i} - t_{v_1,i})F_{v_2}\left(F_{v_1}/(F_{v_1} + F_{v_2})\right).$$

This identifies a better item hand-off location where picker v_1 meets the transporter by traveling toward the transporter for AdditionalTravelDistanceForv_1. Then, the new arrival times for the vehicles are updated via LP relaxation.

Algorithm 8 Pseudocode for the UpdateHandOffLocations() function.

Require: HandOffLocation, PickerRoute, TransporterRoute

1: [ArrivalTime]=LPRelaxation(HandOffLocation,PickerRoute,TransporterRoute)
2: ArrivalTime*=ArrivalTime; HandOffLocation*=HandOffLocation
3: % Determine the hand-off locations one at a time chronologically based on arrival times
4: **for** All task $i \in I^{\mathcal{S}}$ in chronological order **do**
5:     **if** The arrival time $t_{v_1,i}$ of picker v_1 at i is earlier than transporter v_2 **then**
6:         AdditionalTravelDistance=$(t_{v_2,i} - t_{v_1,i})F_{v_2}\ (F_{v_1}/(F_{v_1} + F_{v_2}))$
7:         Origin=Item storage location $i - |I|$
8:         Destination1=The location where v_2 visited prior to i
9:         HandOffLocation'=Updated item hand-off location i in HandOffLocation by
10:            moving i from Origin to Destination1 for AdditionalTravelDistance far
11:         [ArrivalTime]=LPRelaxation(HandOffLocation',PickerRoute,TransporterRoute)
12:         **if** max(ArrivalTime')<max(ArrivalTime*) **then**
13:             ArrivalTime*=ArrivalTime'; HandOffLocation*=HandOffLocation';
14:         **end if**
15:     **end if**
16: **end for**
17: **return** ArrivalTime*,HandOffLocation*

Note that in the construction phase of the proposed heuristic, we set the initial item hand-off locations as the item storage locations. Therefore, the proposed heuristic can be implemented to solve the PPT-VRP by skipping the lines 3–16 in Algorithm 8.

4.5 Numerical analysis

A series of numerical analyses was conducted to (1) assess the efficacy of the proposed heuristic, and (2) quantify the impacts of the APPT-VRP model on order retrieval response time and service level. All computational work was conducted on a PC with a Core Intel i5-2410m processor and 12 GB memory running Microsoft Windows 8 in 64-bit mode. The APPT-VRP models were solved using Gurobi 7.0.2, a popular solver software package. Heuristics were coded in MATLAB, a numerical computing programming package.

4.5.1 Analysis of the APPT-VRP heuristic

Owing to the uniqueness of the APPT-VRP, which we explained in Section 4.2, we had to generate our own test problems as a test-bed to evaluate the effectiveness of the proposed heuristic, and our basis for comparison is to solve the PPT-VRP and APPT-VRP via the MILP solver, which is possible only for small- and mid-scale problem instances. We created 100 randomly generated pick lists with five and ten items, respectively, under a common warehouse layout with six vertical picking aisles and three horizontal cross aisles, as illustrated in Figure 4.1. A uniform storage policy was applied as in previous batch order picking studies (c.f., de Koster and Poort [86], Chew and Tang [84], Roodbergen and de Koster [107, 108], Roodbergen and Vis [109], Ho et al. [95], Nieuwenhuyse and de Koster [103], Parikh and Meller [105], Yu and de Koster [118], Henn [94], Shqair et al. [115], Thomas and Meller [116], Roodbergen et al. [110]). The use of fixed pick list sizes is according to the order picking literature (c.f., de Koster and Poort [86], Roodbergen and de Koster [107, 108], Hwang et al. [97], Roodbergen and Vis [109], Pan et al. [104], Shqair et al. [115]).

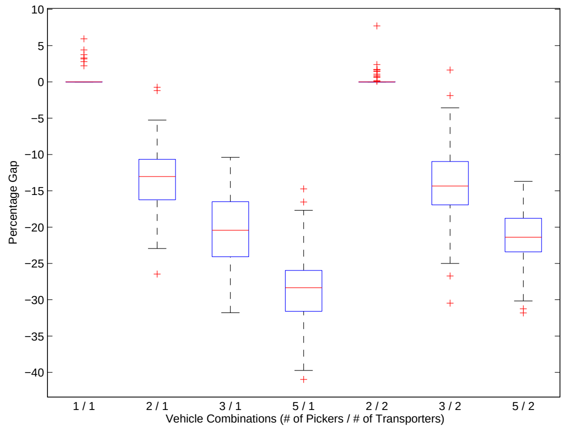
Consistent with Parikh and Meller [105], Pan et al. [104], the widths of cross aisles and picking aisles are 10 feet and 6 feet, respectively, and the size of storage racks is 1-foot wide by 5-feet deep. According to the specifications by [117], both picker robot and transporter robot have a base 22-inches in diameter. Thus, at least three robots can occupy an aisle side-by-side with room to spare. This is why we do not consider aisle blocking in this paper. Seven combinations of pickers and transporters are considered, from one picker and one transporter (1/1), to five pickers and two transporters (5/2). Thus, 1,400 test instances (100 pick lists $\times$ two types of pick list sizes $\times$ seven combinations) are created for the numerical analysis.

Three routing methodologies are evaluated. In the first methodology, the PPT-VRP model from Lee and Murray [100], in which pickers can only hand-off the items to the transporters at the precise item storage location, is applied. In the second, the APPT-VRP model proposed in this paper, which allows the hand-off to happen anywhere in the warehouse, is applied. For five-item problems, Gurobi is able to obtain optimal solutions for the APPT-VRP model within 60 minutes; therefore, we compare the performance of the proposed heuristic and the PPT-VRP model in terms of optimal solutions. For 10-item problems, we set a cut-off time of 30 minutes to obtain the solutions for PPT-VRP and APPT-VRP via Gurobi. The third methodology is the proposed heuristic, and we set a 20-second cut-off time for the proposed heuristic to solve both the five-item and 10-item problems.

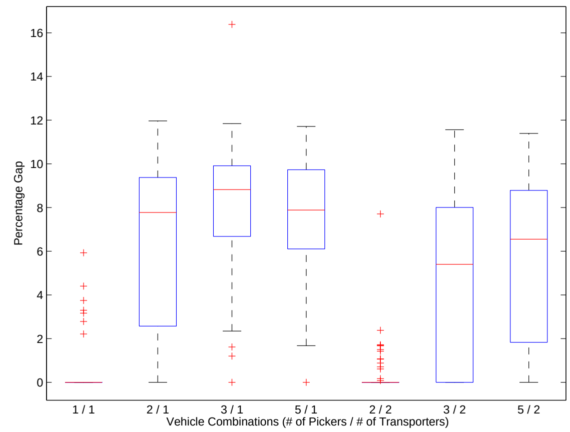
Consistent with Wise et al. [117], pickers traveled at a speed up to 1.0 m/s, while transporters traveled up to 2.0 m/s. Each picker was assumed to require three seconds to pick up or place an item into a tote. We assume the additional travel time required for turning corners is negligible. The tote drop-off time was assumed to be five seconds for transporters. This study focused on warehouse order picking for small items, in which robots are able to operate for hours and each transporter is able to carry more than 10 items on a route. Thus, the capacity of each transporter was set as the pick list size, and we did not consider battery charging related assumptions in the experiment.

Figure 4.3 provides a summary of the APPT-VRP heuristic's performance, where the "gap" reported is the percentage difference from the solutions obtained by the routing methodology to the optimal or best found solutions. For the 10-item problems, using Gurobi with the time limit, none of the APPT-VRP problems could be solved optimally, and the solver even failed to obtain a feasible solution in 86 (out of 700) problems. Additional performance details are provided in Table 4.3.

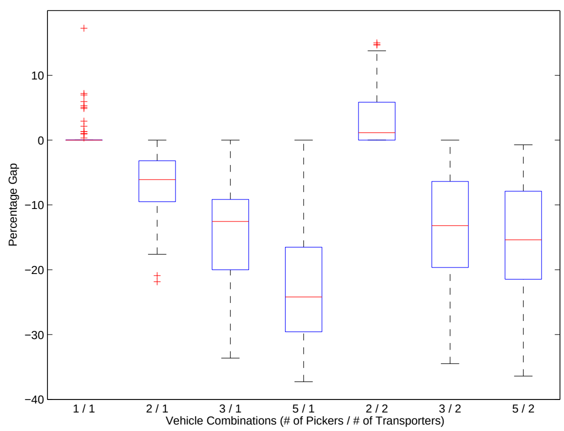
As stated above, the PPT-VRP formulation assumes items hand-off only at the precise item storage locations, but the optimal solutions of the PPT-VRP are equivalent to the APPT-VRP's when the numbers of pickers and transporters are equivalent. In such a case, transporters follow pickers respectively until no more items need to be placed onto transporters. However, PPT-VRP is a more restrictive VRP than the APPT-VRP. Thus, this



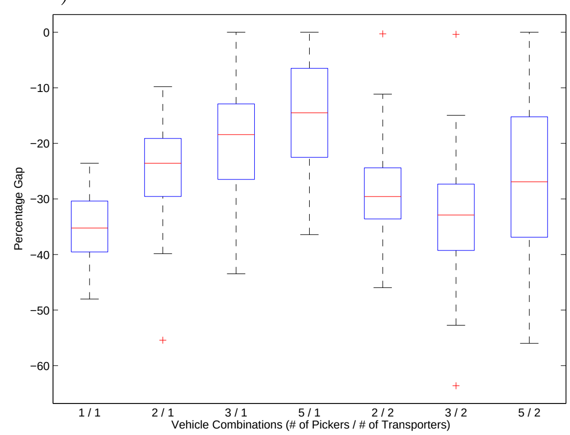
(a) Gap to PPT-VRP optimal solutions (5-item)



(b) Gap to APPT-VRP optimal solutions (5-item)



(c) Gap to 30-min Gurobi solutions of PPT-VRP (10-item)



(d) Gap to 30-min Gurobi solutions of APPT-VRP (10-item)

Figure 4.3: Comparison of the proposed heuristic's effectiveness.

Table 4.3: Details of heuristic performance.

Solution Approach	**Pick List Size**	**Opt./Best Found Gap in Avg.** (%)	**Opt./Best Found Sol. Obtained** (%)	**Runtime** (s)		**Makespan + Runtime in Avg.** (s)
				Avg	**Max**	
Proposed heuristic	5	4.7	36.7	20	20	79.4
PPT-VRP IP		28.5	28.6	1.1	4.4	71.1
APPT-VRP IP (20s)		10.6	9.6	19.4	20.0	83.6
APPT-VRP IP (Opt.)		0.0	100.0	233.6	3020.0	292.2
Proposed heuristic	10	0.8	87.0	20.0	20.0	107.7
PPT-VRP IP (30min)		13.2	29.1	1143.9	1800.0	1239.8
APPT-VRP IP (30min)		36.9	2.7	1800.0	1800.0	1909.2

analysis compares the solution of these two problems to demonstrate how much improvement on makespan by relaxing the assumption of PPT-VRP, which is the picker can only do the hand-off at the very item storage locations. Although the PPT-VRP with 5 items only took Gurobi a second on average to solve optimally, negative gaps, which mean the proposed heuristic is able to obtain better solutions than PPT-VRP's optimal solutions, were reported for the problems with more pickers than transporters. In the 10-item problems, the average gaps for all the vehicle combinations do not change significantly, but it took approximately 20 minutes to solve. Thus, for the problems with more pickers than transporters, the proposed heuristic is superior to the solutions of the PPT-VRP model obtained by Gurobi.

The proposed APPT-VRP heuristic is able to obtain near-optimal solutions with less than 10% gap in 20 seconds for most of the test problems, which took Gurobi four minutes on average. Moreover, the solutions obtained by Gurobi in 20 seconds are not sufficiently competitive. Thus, considering that the optimal makespan for most of the five-item test problems is less than four minutes, the proposed heuristic demonstrates its own value in practice. Moreover, the proposed heuristic is able to solve more than 30% of the test problems optimally in 20 seconds, whereas PPT-VRP is only able to obtain optimal solutions when the number of pickers is equal to that of transporters in the system.

We also provide the average total required time to fulfill a pick list by using the three methodologies, including the computational time to obtain a routing plan plus the actual required time for the robots to retrieve all items from the pick list. This is because in practice, the customers need to wait on not only the time to retrieve the orders but also the computational times to obtain the routing plans since their orders have been placed. Thus, a reasonable computational time becomes more important in terms of optimizing the overall service level. In our experiment, the solutions obtained by the proposed heuristic dominates the solutions of the two formulation models obtained by Gurobi in the average total required time for both 5-item and 10-item problems, except the PPT-VRP in 5-item problems. That is because although better solutions could be obtained by the proposed

Table 4.4: Details of heuristic performance:5 item problems from a 10 times larger warehouse.

Solution Approach	**Opt./Best Found Gap in Avg.** (%)	**Opt./Best Found Sol. Obtained** (%)	**Runtime (s)**		**Makespan + Runtime in Avg.** (s)
			Avg	**Max**	
Proposed heuristic	4.7	36.7	20.0	20.0	470.7
PPT-VRP IP	28.5	28.6	1.7	16.1	562.6

heuristic with identifying better item hand-off locations for 5-item problems, the saving is not significant enough than the solutions of the PPT-VRP due to the size of the warehouse from the literature. Although the warehouse configuration is from the literature, it is relatively small compared to the warehouses in real world. For this reason, we compare the proposed heuristic and the PPT-VRP model in solving the same 5-item problems under a same shape, but ten times large, more practical size warehouse. The information of the comparison is provided in Table 4.4, which shows that the required makespan increases approximately ten times longer under the warehouse, but the required computational times do not change much for both methodologies. Therefore, our heuristic dominates the PPT-VRP model in total required time under practical size warehouses.

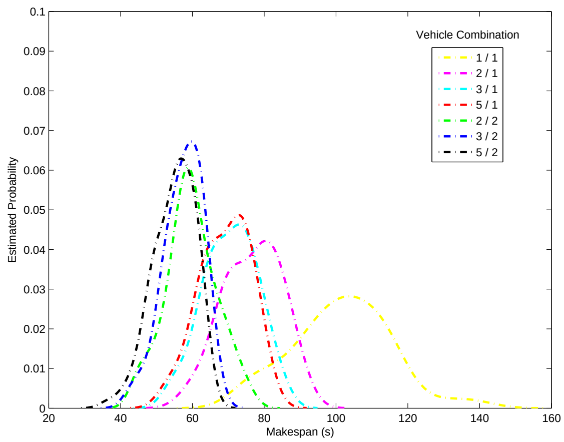
A warehouse receives orders continuously from different customers and needs to fulfill orders as soon as possible. Thus, the runtime for computing the routes should be considered in the total completion time, which is from the time a pick list order is received until the time that the pick list is fulfilled. Therefore, we provide the average total completion time, which includes the average runtime plus the average makespan, between receiving a pick list and fulfillment of that list. This signifies the importance of having an efficient heuristic that is able to provide optimal or near-optimal solutions in a short time. Instead of running Gurobi until all the optimal solutions are obtained, it is more practical to use the proposed heuristic considering that most of the pick lists can be fulfilled within the runtime required to solve APPT-VRP optimally. Although the PPT-VRP formulation is able to obtain a slightly shorter total average completion time for five-item problems, it is not comparable for mid- or larger-scale problems.

As demonstrated in this section, the numerical analysis shows that high-quality (i.e., near-optimal) solutions can be obtained via the proposed heuristic. In the next section, owing to the randomness of the orders under uniform storage policy, we investigate the difference between the two formulations and among different vehicle combinations in terms of overall makespan and service level.

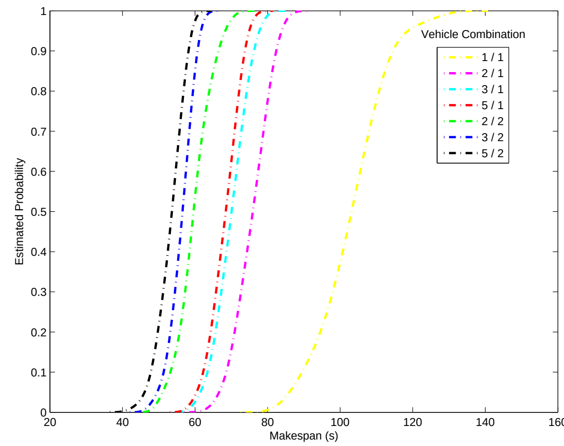
4.5.2 PPT-VRP versus APPT-VRP

The above analysis explored the effectiveness of the proposed heuristic. In this section, we identify the overall response time and service level of the two problem classes. In other words, we are interested in the overall performance of the two formulations in terms of the makespan for random orders under a uniform storage policy. Instead of focusing on the average response time, we are interested in the variance and the distribution of the makespan among the formulations and the vehicle combinations.

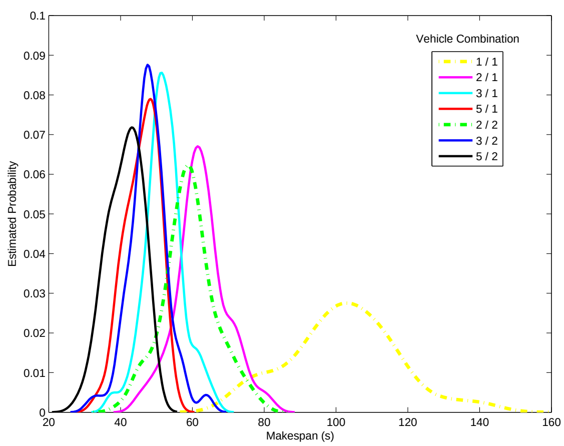
We analyzed the five-item optimal results of the two formulations obtained by Gurobi from the previous experiment. Thus, the experimental details in this numerical analysis are the same as in the previous section. Figure 4.4 illustrates the estimated distributions of the makespan for both models with different combinations. The probability density functions (PDFs) represent the estimated probability of the required makespan to retrieve the entire pick list, and the cumulative distribution functions represent the cumulative probability. For example, in the PDF plots, it is clear that there is a 3% probability that the makespan is 105 seconds for one picker and one transporter, and in the CDF plots, 90% of the orders can be retrieved in 40 (58) seconds for the APPT-VRP (PPT-VRP) model when there are five pickers and two transporters. This analysis provides a more comprehensive means of comparing different warehousing systems in terms of the overall service level. Note that both models have the same optimal solutions when the number of pickers is the same as that of transporters, as explained above.



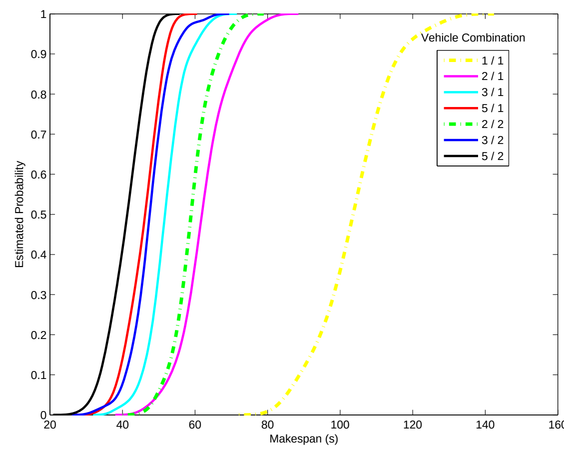
(a) Estimated probability density function of makespan for PPT-VRP



(b) Estimated cumulative distribution function of makespan for PPT-VRP



(c) Estimated probability density function of makespan for APPT-VRP



(d) Estimated cumulative distribution function of makespan for APPT-VRP

Figure 4.4: Improvement of makespan / service level by obtained using the APPT-VRP model.

As shown in the figure, there was a marked difference between the PPT-VRP and the APPT-VRP in the distributions. In particular, when there are more pickers than transporters, distributions in the PPT-VRP model are relatively flatter and wider than the APPT-VRP's. Consistent with our previous study, which indicated the importance of balancing the numbers of pickers and transporters for the PPT-VRP model, having more pickers than transporters does not significantly improve the results. However, having either a ratio of three (pickers):one (transporter) or 5:1 dominates 2:2 not only in terms of average makespan but also in terms of variances for the APPT-VRP model.

Table 4.5: Statistical comparison of APPT-VRP and PPT-VRP in makespan (s).

Formulation	Descriptive Statistics	Vehicle Combination (Picker/Transporter)						
		1/1	**2/1**	**3/1**	**5/1**	**2/2**	**3/2**	**5/2**
APPT-VRP	Avg	102.0	62.4	52.0	45.9	59.0	47.6	41.1
	Var	191.8	41.7	22.8	18.9	45.9	23.1	19.7
	Range	68.2	33.0	26.8	20.0	32.2	32.40	18.5
	95% CI width	1.9	0.9	0.6	0.6	0.9	0.7	0.6
PPT-VRP	Avg	102.0	76.9	70.8	69.3	59.0	57.5	55.0
	Var	191.8	57.3	49.3	43.3	45.9	26.6	30.1
	Range	68.2	33.5	31.5	28.5	32.2	23.0	24.0
	95% CI width	1.9	1.0	1.0	0.9	0.9	0.7	0.7

An interesting observation is that a small portion of the orders could be fulfilled faster with a ratio of 3:2 than 5:1. These orders are stored relatively close to the depot, therefore having more transporters enables the system to take advantage of the shorter required traveling distance and do the hand-offs simultaneously. Otherwise, a ratio of 5:1 is better than 3:2 for most of the random orders in this case. Table 4.5 shows that the variance of makespan for the PPT-VRP model is larger than the corresponding APPT-VRP model with the same vehicle combinations when there are more pickers than transporters. This means that with the APPT-VRP model, the possibility of missing the shipping deadline is less in practice. Moreover, statistically, we are confident of our experiment's accuracy because the 95% confidence intervals (CIs) are relatively narrow. The analysis demonstrates the practical value of APPT-VRP. An attractive future research topic would be to explore the performance of the APPT-VRP formulation under different storage policies.

4.6 Conclusions and future research opportunities

As an extension of [100], the APPT-VRP was proposed to improve the overall service level by relaxing the assumptions of item hand-off locations. An MILP formulation of this problem was presented to verify the improvements obtained via the relaxation. Owing to the problem being NP-hard, only small-scale problems may be solved optimally via MILP solvers. Moreover, even for small-scale problems, the excessive computation time required by the existing MILP solver is longer than the actual order retrieval time. Therefore, an efficient heuristic was proposed to solve the problem in a reasonable time for practice. The solution quality of the proposed heuristic was validated via numerical analysis for small- and mid-scale problems.

Several future research opportunities are listed. First of all, although this work considered the order retrieval for given fixed size pick lists, random size customer orders are received over time in real-world continuous warehouse operation. Thus, the dynamic routing problem for the continuous environment is attractive. In such a case, another interesting topic would be to determine suitable order-batching methodologies for robot-based warehousing. Additionally, the re-stocking activities by the warehousing robotics, in which items flow from the depot to storage shelves by the robotics, could be an interesting research topic.

[83] D. Bredstrom and M. Ronnqvist. Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *European Journal of Operational Research*, 191 (1):19–31, 2008.

[84] E.P. Chew and L.C. Tang. Travel time analysis for general item location assignment in a rectangular warehouse. *European Journal of Operational Research*, 112(3):582–597, 1999.

[85] L.C. Davis. Dynamic origin-to-destination routing of wirelessly connected, autonomous vehicles on a congested network. *Physica A: Statistical Mechanics and its Applications*, 478: 93–102, 2017.

[86] R. de Koster and E.V.D. Poort. Routing orderpickers in a warehouse: A comparison between optimal and heuristic solutions. *IIE Transactions*, 30(5):469–480, 1998.

[87] M. Drexl. Synchronization in vehicle routing: A survey of VRPs with multiple synchronization constraints. *Transportation Science*, 46(3):297–316, 2012.

[88] M. Drexl. Branch-and-cut algorithms for the vehicle routing problem with trailers and transshipments. *Networks*, 63(1):119–133, 2014.

[89] B.Y. Ekren, S.S. Heragu, A. Krishnamurthy, and C.J. Malmborg. Simulation based experimental design to identify factors affecting performance of AVS/RS. *Computers & Industrial Engineering*, 58(1):175–185, 2010.

[90] A. Ferrara, E. Gebennini, and A. Grassi. Fleet sizing of laser guided vehicles and pallet shuttles in automated warehouses. *International Journal of Production Economics*, 157:7–14, 2014.

[91] fetchrobotics, 2017. URL `http://fetchrobotics.com/`. Last visited on 05/26/2017.

[92] M. Fukunari and C.J. Malmborg. A network queuing approach for evaluation of performance measures in autonomous vehicle storage and retrieval systems. *European Journal of Operational Research*, 193(1):152–167, 2009.

[93] S.R. Ait Haddadene, N. Labadie, and C. Prodhon. A GRASP $\times$ ILS for the vehicle routing problem with time windows, synchronization and precedence constraints. *Expert Systems with Applications*, 66:274–294, 2016.

[94] S. Henn. Algorithms for on-line order batching in an order picking warehouse. *Computers & Operations Research*, 39(11):2549–2563, 2012.

[95] Y.C. Ho, T.S. Su, and Z.B. Shi. Order-batching methods for an order-picking warehouse with two cross aisles. *Computers & Industrial Engineering*, 55(2):321–347, 2008.

[96] W. Hu, J. Mao, and K. Wei. Energy-efficient rail guided vehicle routing for two-sided loading/unloading automated freight handling system. *European Journal of Operational Research*, 258(3):943–957, 2017.

[97] H. Hwang, Y.H. Oh, and Y.K. Lee. An evaluation of routing policies for order-picking operations in low-level picker-to-part system. *International Journal of Production Research*, 42(18):3873–3889, 2004.

[98] IAMrobotics, 2017. URL `http://www.iamrobotics.com/`. Last visited on 05/26/2017.

[99] P. Kuo, A. Krishnamurthy, and C.J. Malmborg. Design models for unit load storage and retrieval systems using autonomous vehicle technology and resource conserving storage and dwell point policies. *Applied Mathematical Modelling*, 31(10):2332–2346, 2007.

[100] H. Lee and C.C. Murray. Robotics in order picking: Evaluating warehouse layouts for pick, place, and transport vehicle routing systems. *International Journal of Production Research (Under Review)*, 2017.

[101] Locusrobotics, 2017. URL `http://www.locusrobotics.com/`. Last visited on 05/26/2017.

[102] C.C. Murray and A.G. Chu. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54: 86 – 109, 2015.

[103] I.V. Nieuwenhuyse and R. de Koster. Evaluating order throughput time in 2-block warehouses with time window batching. *International Journal of Production Economics*, 121(2):654–664, 2009.

[104] J.C. Pan, M. Wu, and W. Chang. A travel time estimation model for a high-level picker-to-part system with class-based storage policies. *European Journal of Operational Research*, 237 (3):1054–1066, 2014.

[105] P.J. Parikh and R.D. Meller. A travel-time model for a person-onboard order picking system. *European Journal of Operational Research*, 200(2):385–394, 2010.

[106] R. Redjem and E. Marcon. Operations management in the home care services: a heuristic for the caregivers' routing problem. *Flexible Services and Manufacturing Journal*, 28(1):280–303, 2016.

[107] K.J. Roodbergen and R. de Koster. Routing order pickers in a warehouse with a middle aisle. *European Journal of Operational Research*, 133(1):32–43, 2001.

[108] K.J. Roodbergen and R. de Koster. Routing methods for warehouses with multiple cross aisles. *International Journal of Production Research*, 39(9):1865–1883, 2001.

[109] K.J. Roodbergen and I.F.A. Vis. A model for warehouse layout. *IIE Transactions*, 38(10): 799–811, 2006.

[110] K.J. Roodbergen, I.F.A. Vis, and G.D. Taylor. Simultaneous determination of warehouse layout and control policies. *International Journal of Production Research*, 53(11):3306–3326, 2015.

[111] L. Rousseau, M. Gendreau, and G. Pesant. The synchronized dynamic vehicle dispatching problem. *INFOR: Information Systems and Operational Research*, 51(2):76–83, 2013.

[112] D. Roy, A. Krishnamurthy, S.S. Heragu, and C.J. Malmborg. Performance analysis and design trade-offs in warehouses with autonomous vehicle technology. *IIE Transactions*, 44 (12):1045–1060, 2012.

[113] D. Roy, A. Krishnamurthy, S. Heragu, and C. Malmborg. Queuing models to analyze dwell-point and cross-aisle location in autonomous vehicle-based warehouse systems. *European Journal of Operational Research*, 242(1):72–87, 2015.

[114] M. Saidi-Mehrabad, S. Dehnavi-Arani, F. Evazabadian, and V. Mahmoodian. An ant colony algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs. *Computers & Industrial Engineering*, 86:2–13, 2015.

[115] M. Shqair, S. Altarazi, and S. Al-Shihabi. A statistical study employing agent-based modeling to estimate the effects of different warehouse parameters on the distance traveled in warehouses. *Simulation Modelling Practice and Theory*, 49:122–135, 2014.

[116] L.M. Thomas and R.D. Meller. Developing design guidelines for a case-picking warehouse. *International Journal of Production Economics*, 170, Part C:741–762, 2015.

[117] M. Wise, M. Ferguson, D. King, E. Diehr, and D. Dymesich. Fetch & Freight: Standard platforms for service robot applications. In *The IJCAI (International Joint Conference on Artificial Intelligence)-2016 Workshop on Autonomous Mobile Service Robots*, 2016.

[118] M. Yu and R. de Koster. Enhancing performance in order picking processes by dynamic storage systems. *International Journal of Production Research*, 48(16):4785–4806, 2010.

Chapter 5

Conclusions

This research explores the intersection of facility layout and vehicle routing problems. The facility problem, described in Chapter 2, focused on the FLP with space utilization concerns for semiconductor manufacturing fabs to conduct the facility layout alternatives to satisfy future demand. The space utilization, which is more important for semiconductor manufacturing, is considered as the objective. The novel FLP, which considers facility layout planning in both FA and machine level arrangements, is defined and formulated with practical constraints. Due to the scale of the practical size problems, an efficient heuristic is proposed. In order to demonstrate the quality of the solutions obtained by the proposed heuristic, a better lower bound, which is also computationally accessible, is proposed as well. The numerical analysis demonstrates the proposed heuristic is able to obtain solutions with single-percentage gaps to the proposed lower bound, meaning the optimality gaps are smaller. This indicates the feasibility of fitting all required machines into a constrained facility can be determined by the proposed approach. As a result, the proposed approach helps to provide realistic information for future production planning while building or re-modeling a fab. Furthermore, given machine sets generated based on possible future demands, the facility layout solutions can support business decision-making in terms of long-term supply commitments to the customers.

Served as a bridge between the FLP and the VRP, Chapter 3 proposed a novel VRP for autonomous warehousing systems and studied the warehouse layout alternative analysis when the proposed routing system is employed. A novel depot type, CDs, for warehouses is presented. Several managerial insights have been presented based on the results of the numerical analysis. First, from the perspective of layout planning, CDs outperform TDs because ADBPL will be shorter when CDs are adopted. Applying CDs in the warehouses also eases the negative impact of adding CAs. Second, from the perspective of vehicle capability and combinations, increasing the number (pick-up speed) of pickers helps better when having more transporters than pickers and under warehouses with high (low) ADBPL. Increasing the number (payload capacity) of transporters is preferable when having more pickers than transporters and under the warehouses with low (high) ADFD. Increasing the capability of transporters (pick-up speed of pickers) is more effective when having sufficient numbers of pickers and transporters, and under the warehouses with high (low) ADFD. Last but not least, increasing the pick-up speed is preferable when having sufficient numbers of pickers and high-capacity transporters or under the warehouses with low ADFD and ADBPLS, since there is no much room for the improvement by adding more vehicles to share the travel workloads. This work provides a foundation for a variety of future research opportunities. Due to the NP-hardness of the PPT-VRP, large-scale pick lists were not considered in the analysis of optimal solutions in this study. Therefore, efficient heuristics for solving the large-scale problems are desirable. The analysis also provides a guideline for the future development of the warehousing robotics.

As an extension of Chapter 3, Chapter 4 presented the APPT-VRP with relaxing the assumptions of item hand-off locations in the PPT-VRP. Due to the excessive computational time required to solve the routing problem via commercial MILP solver in our preliminary results, an efficient heuristic is proposed to solve the routing problems for the warehousing robotics. We demonstrated not only the quality of the solutions obtained by the proposed heuristic, but also the improvement of overall warehouse service level by the relaxation.

Future research efforts might be directed in two areas: (1) extending the FLP and the VRPs with considering material handling flow assignment problems, and (2) the studies of the autonomous warehousing systems under continuous operation. As introduced in Chen et al. [6], the capacity planning process in semiconductor manufacturing determines the number of required machines based on the estimates of the material handling distances among FAs when functional area layouts are adopted. However, the actual distances of the material handling flows are depended on the results of the facility layout planning. Moreover, it's possible to have more than one process flows within a FA to produce a type of products in practice. Therefore, to optimize the FLPs with considering material handling flow assignment problems can be beneficial in terms of not only shortening the material handling distances but also reducing the errors caused by the estimation in the current capacity planning process.

This research introduced the warehouse layout alternative analysis for the warehousing robotics in Chapter 3 under the uniform storage policy as in the literature. However, different storage policies are applied in practice, depending on the warehouse storage capacities and the customer demands. Adopting different storage policies results in the changes of ADFDs and ADBPLs, which directly impact the performance of the warehousing systems. Therefore, it would be interesting to optimize the warehouse layout designs with the consideration of where items should be stored when the autonomous systems are implemented.

Although the majority of our focus is on the deterministic VRPs for the warehousing robotics, the dynamic routing problems and methodologies for continuous warehouse operation when orders are received over time could be interesting. In such case, if the turnover rates are high, the potential objective functions of the dynamic problems could be minimizing the number of delayed orders given different shipping due time for each order or maximizing the number of fulfilled orders in a given time period. Thus, performance evaluation models for the warehousing systems under different operational strategies are also attractive. Moreover, the available computational times for solving the dynamic routing problems can be more restricted. Additionally, if the turnover rates are relatively low, where the idle robots should stand-by may impact the warehouse service level. Therefore, the studies of evaluating different zoning strategies for the robots are desirable.

[6] J.C. Chen, R. Dai, and C. Chen. A practical fab design procedure for wafer fabrication plants. *International Journal of Production Research*, 46(10):2565–2588, 2008.