COLLABORATIVE HUMAN-ROBOT ORDER PICKING SYSTEM: ALGORITHMS FOR TASK ALLOCATION AND ROUTING IN COMPLEX ENVIRONMENTS

A Dissertation

presented to

the Faculty of the Graduate School

at the University of Missouri-Columbia

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy in Industrial Engineering

---

by

Shitao Yu

Dr. Sharan Srinivas, Dissertation Supervisor

July 2022

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled

COLLABORATIVE HUMAN-ROBOT ORDER PICKING SYSTEM: ALGORITHMS FOR TASK ALLOCATION AND ROUTING IN COMPLEX ENVIRONMENTS

presented by Shitao Yu, a candidate for the degree of Doctor of Philosophy in Industrial Engineering, and hereby certify that in their opinion it is worthy of acceptance.

Dr. Sharan Srinivas

Dr. James Noble

Dr. Suchi Rajendran

Dr. Aldis Jakubovskis

Dedicated to my parents, Xudong and Jiling, for their support and inspiration throughout my journey.

Acknowledgement

I would like to express my heartfelt gratitude to the individuals who have provided me with their unwavering encouragement, support, love, and valuable advice throughout the course of this research. It is through their direct and indirect assistance that I have been able to successfully complete this dissertation.

First and foremost, I would like to express my deepest gratitude to the peaceful relation between the two countries, as it has paved the way for me to pursue my studies abroad for a long period of time in pursuit of my degree. Next, I would like to thank Dr. Srinivas Sharan, my adviser, for his support, guidance, encouragement and technical expertise under which I was able to perform in-depth research. Under his invaluable guidance, I had the privilege of acquiring the skills and qualities that define an exceptional human being and an effective leader. His unwavering help and guidance were instrumental in enabling me to attain my objectives successfully.

I thank my dissertation committee members, Dr. Noble James, Dr. Rajendran Suchithra, and Dr. Jakubovskis Aldis, for providing continual feedback and for serving on my committee. Special thanks to Dr. Noble for serving as my master's degree advisor, providing invaluable guidance and support throughout my research journey. I am extremely grateful to the Department of Industrial

and System Engineering at University of Missouri for awarding me the fellowship, allowing me to spend more time on research with no financial consequences. Special thanks to Dr. Rajendran and Dr. Jakubovskis for their exceptional guidance and expertise in my academic courses.

I am deeply grateful to my parents, Yu Xudong and Yang Jiling, for their unwavering love, support, and guidance throughout my academic journey. Their teachings of honesty and hard work have shaped me into the person I am today. I am particularly grateful for their encouragement and belief in my decision to study abroad, and their unwavering pride in my accomplishments fills me with joy and gratitude. And I would like to thank all my extended family members for their love and affection.

Table of Contents

List of Figures

List of Tables

COLLABORATIVE HUMAN-ROBOT ORDER PICKING SYSTEM: ALGORITHMS FOR TASK ALLOCATION AND ROUTING IN COMPLEX ENVIRONMENTS

Shitao Yu

Dr. Sharan Srinivas, Dissertation Supervisor

Abstract

Order picking, which involves retrieving items from storage locations for an internal or external customer, is a core function in warehouses and accounts for up to 65% of the total operating cost. It is considered as a crucial driver for supply chain performance as improper planning of picking operations leads to inefficient asset utilization and delayed deliveries, which, in turn, adversely affects customer satisfaction, operating cost, and competitiveness. A majority of warehouses adopt a picker-to-parts system (workers travel through a warehouse to retrieve and transport items from storage to packing station), and the fulfillment speed of such order picking system (OPS) depends on the following key decisions - (i) set of orders to be picked in a tour (order batching decision), (ii) assignment of a batch to a picker and order in which their a processed (batch assignment and sequencing decisions), and (iii) route followed

by the picker to collect the orders in each batch (picker routing decision). However, with the growing customer demand and global labor shortage, warehouses are seeking efficient and less labor-intensive order picking systems. Autonomous mobile robots (AMRs) or collaborative robots (cobots) have the potential to alleviate the strain on human workers and expedite order picking operations. However, there are several key operational challenges to address for ensuring an effective collaborative human-robot order-picking system (CHR-OPS), where humans perform item retrieval tasks, and AMRs handle item transportation to the depot.

This research aims to improve the fulfillment efficiency of a picker-to-parts CHR-OPS by optimizing key decisions associated with two warehouse picking strategies, namely, static picking and dynamic picking. In the case of static picking, the items to be retrieved from storage for a given day are known as apriority. On the other hand, the dynamic picking strategy allows for orders to arrive over time (e.g., e-commerce warehouses), and the pick cycle (or picking plan) can be updated in real time.

First, we address the problem of optimizing the following key subproblems of a CHR-OPS with a static picking strategy: (i) order batching (how many items should be collected in each AMR tour?), (ii) batch assignment and sequencing (how to assign batches to AMRs, and in what order should they be processed?), and (iii) picker-robot routing (how should the AMR and picker be routed to coordinate the picking process?). Existing literature has not dealt with the three subproblems, and this work

is the first to address them for a picker-to-parts CHR-OPS system employing multiple pickers and AMRs. A mixed integer linear programming model is developed to jointly optimize the three subproblems with the objective of minimizing the total tardiness of all orders. The MILP model is validated and solved to optimality for small instances. However, since the problem under consideration is NP-hard, it is computationally intractable for larger instances. To efficiently handle large instances, we proposed deterministic and stochastic local search algorithms, namely variable neighborhood descent and a new variant of simulated annealing. The numerical experiments demonstrate the superior performance of the proposed solution approach compared to existing methods. Besides, our results also show that the picking efficiency is impacted by human–robot team composition, AMR speed, AMR capacity and warehouse layout.

Subsequently, we address the CHR-OPS with dynamic picking and developing interventionist picking algorithms for a multi-robot multi-human setting. Specifically, we propose two interventionist policies for dynamic collaborative order picking, namely, the collaborative human-robot interventionist picking algorithm (CHR-IPA) and the collaborative human-robot rule-based interventionist picking algorithm (CHR-RIPA). The evaluation of the policies demonstrates that the proposed rules can improve the overall performance of the system compared to benchmark approaches (human-only dynamic picking and collaborative picking with no intervention). In addition, results indicate CHR-IPA outperforms CHR-RIPA in terms of average tar-

diness and order completion time, albeit with a slightly higher travel distance for human workers.

The results have led to several managerial implications for a collaborative order picking system. Further, the proposed models and algorithms are modular and can be adapted to any warehouse setting by accounting for the relevant parameters such as warehouse layout, AMR capacity and human-robot composition. Finally, the directions for future research have been identified and summarized.

Chapter 1

Introduction

Order picking, which involves retrieving items from storage locations for an internal or external customer, is a core function in warehouses and accounts for up to 65% of the total operating cost (Ho et al., 2007). Moreover, order picking is considered as a crucial driver for supply chain performance as improper planning of picking operations leads to inefficient asset utilization and delayed deliveries, which, in turn, adversely affects customer satisfaction, operating cost, and competitiveness (Scholz et al., 2017). A majority of the warehouses adopt the traditional picker-to-parts order picking system (OPS), where pickers travel through the warehouse to retrieve and transport the items from their storage location to the packing station. For instance, 80% of warehouses in Western Europe employ a picker-to-parts OPS (de Koster et al., 2007; Napolitano, 2012). Such warehouses typically use low-level

picking by storing the stock keeping units (SKUs) or items in storage racks. With the recent rapid growth in the e-commerce sector and changing consumer delivery expectations, warehouses are now faced with the challenge of handling high demand and tight due dates.

The fulfillment speed of an OPS is dependent on the following key decisions – (i) set of orders to be picked in a tour (order batching decision), (ii) assignment of a batch to a picker and order in which their a processed (batch assignment and sequencing decisions), and (iii) route followed by the picker to collect the orders in each batch (picker routing decision). While the picker-to-parts system requires a relatively low investment cost, it is labor-intensive and contributes to about 60% of all labor activities in the supply chain.

1.1 Background and Challenges

Order picking strategies in warehouses can be categorized as static or dynamic. Static order picking involves fixed pick lists that are predetermined and remain unchanged throughout the picking process. This strategy is suitable for scenarios with stable demand and limited variability, such as traditional retail stores. In contrast, dynamic order picking allows for adjustments and modifications to the pick list based on changing demand. It is particularly valuable in dynamic environments like e-

commerce warehouses, where orders are unknown in the beginning and can arrive anytime dynamically. Dynamic order picking enhances flexibility and responsiveness, enabling warehouses to efficiently handle the high variability and uncertainty associated with e-commerce fulfillment.

Automation, enabled by Industry 4.0 technologies, provides the capability to reduce the dependence on labor resources and improve overall warehouse operations (Caputo & Pelagagge, 2006). In particular, autonomous mobile robots (AMR) can alleviate the strain on human pickers as it provides the following advantages.

- *Flexibility and Safety*: AMRs can navigate freely (without being restricted to a pre-determined path or operator supervision) and safely around humans in a warehouse environment using sensors and onboard computers. In particular, AMRs are well-suited to traverse narrow aisles and congested areas. Thus, these features allow them to transport SKUs within a warehouse, thereby reducing the walking distance of humans.
- *Faster and Efficient Transport*: A warehouse robot can carry higher payload capacity and travel faster than a human. Besides, AMRs provide precise fatigue-free operation, thereby rendering them suitable for continuous usage.
- *Easy Integration and Scalability*: AMRs do not require any sophisticated infrastructure to be implemented, and therefore can be easily integrated into the warehouse management system, which, in turn, offers the capability to scale up

or down depending on the demand levels.

While the above-mentioned capabilities make AMRs ideal for transport, their gripping/grasping ability is inferior to human grasping. This is mainly because robotic grasping of an object in a dynamic environment is limited by the current technology such as 3D image identification, sensor and pressure feedback mechanism, and object recognition availability (Lee & Murray, 2019). While specialized picker robots may be used to complement the AMRs, such a fully automated OPS becomes a capital-intensive operation (Drury, 1988; Goetschalckx & Ashayer, 1989; Tompkins et al., 2010), as opposed to a traditional setup. Rational division of picking and carrying between robot and human enables to give full play to their respective strengths. Thus, a collaborative human-robot order picking system (CHR-OPS), where humans perform item retrieval task (picking from storage and placing them on robots) and AMRs transports these items to the drop-off location, has the potential to leverage the strengths of both the resources (humans and AMRs) to achieve faster fulfillment.

The CHR-OPS can be broadly categorized into follow-pick (picker-in lead or robot-in lead) and swarm systems. The former category employs AMR that accompanies human workers performing the pick-and-place operation and returns independently to the packing station once all the items in a batch have been collected, reducing the walking distance and cart operation effort for humans. The routes adopted by each human-robot pair are very similar in the follow-pick system because the AMR either

follows the picker (picker-in lead) or leads the human worker (robot-in lead). Such a system is well-suited for transitioning from a single order picking method (fulfill one order at a time) to a batch order picking system (bulk retrieval of SKUs to deliver multiple orders per tour). Nevertheless, the follow-pick system may not achieve the best trade-off between fulfillment speed and resources employed (pickers and humans) for a warehouse characterized low density, high volume pick environment. On the other hand, the swarm system is applicable to different warehouse environments. It uses AMRs that navigate independently and allows any nearby human picker to meet with a robot to perform the pick-and-place task. However, unlike the follow-pick system, the routes of the multiple AMRs and human pickers need not be similar and must be efficiently planned or optimized. Figure 1.1 shows the two types of robots used in the follow-pick and swarm CHR-OPS.

(a) (b)

Figure 1.1: Illustrative representation of autonomous mobile robots employed in the (a) follow-pick [source: www.dhl.com] and (b) swarm systems [source: https://6river.com]

There are critical subproblems affecting the fulfillment speed for a CHR everonment - order batching, batch assignent, sequencing and robot-picker routing (CHR-OBASRP). Unlike athe traditional human-only OPS, the CHR-OBASRP should handle the coordinated routing of two resource types (AMRs and pickers) so that the item hand-off from the picker to the AMR happens at the pick location. In other words, the route plan must ensure that each item location is visited exactly once by a human-robot pair, while ensuring that both resource types move from the pick location only after the item is placed on the AMR.

1.2 Research Objectives and Contributions

Motivated by the potential of AMRs and the challenges faced by the warehouses (tight deadlines, high volume orders), this work seeks to develop a CHR-OPS to optimize fulfillment efficiency and responsiveness associated with the pick operations by utilizing multiple swarming AMRs and human pickers. Specifically, this research contributes to the literature on order picking in the following ways:

- Introduces a swarm-type CHR-OPS, where multiple swarming robots and human pickers are working in cooperation to retrieve and transport the items pertaining to different orders. To the best of the authors' knowledge, this is the first research to simultaneously consider the critical problems affecting the fulfillment

speed for a collaborative human–robot environment — order batching, batch assignment and sequencing, and robotpicker routing, which is hereafter referred to as CHR-OBASRP. Unlike the traditional human-only OPS, the CHR-OBASRP will handle the coordinated routing of two resource types (AMRs and pickers) so that the item hand-off from the picker to the AMR happens at the pick location. In other words, the route plan must ensure that each item location is visited exactly once by a human–robot pair, while ensuring that both resource types move from the pick location only after the item is placed on the AMR.

- In Chapters 3-5, unlike most prior works (Žulj et al., 2021; Lee and Murray, 2019), we allow multiple items belonging to the same order to be split among different resources, thereby providing the flexibility to improve the picking efficiency. In practice, the items associated with an order may be stored at far-way locations depending on the storage strategy (random, demand-based). Thus, restricting the order from being split among multiple pickers will lead to low pick density, thereby delaying order delivery.

- In Chapter 3, a new mixed integer linear programming (MILP) model is developed and validated for the proposed CHR-OBASRP.

- The traditional order batching, assignment and sequencing, and routing problem (OBASRP) is known to be NP-hard (Scholz et al., 2017). Since this work extends OBASRP to a collaborative human–robot environment, it is also NP-hard. To handle the computational complexity associated with CHR-OBASRP

for large problem instances, in Chapters 3-5, we develop metaheuristic-based solution approaches. Specifically, we propose a restarted simulated annealing algorithm with an adaptive neighborhood search (RSA-ANS) mechanism to better explore and exploit large search space. The proposed approach integrates the SA framework and acceptance criterion with an adaptive selection of neighborhood structures, thereby allowing the exploration of multiple neighborhood solutions at each iteration. In addition, an optimization-based restart strategy is considered to escape from local optima.

- In Chapters 3-5, we introduce various test instances to evaluate the CHR-OBASRP. First, small problem instances are created to validate the proposed MILP model, and evaluate the effectiveness of the RSA-ANS approach. Further, the impact of human–robot composition, AMR speed, AMR cart capacity and warehouse layout is also studied.

- In Chapter 6, to the best of our knowledge, we are the first to introduce interventionist strategies to CHR-OPS to handle dynamic customer orders. We also introduce two variants to the proposed interventionist approach based on different assignment policies.

- In Chapter 6, we propose rule-based approach to modify pick list after the ADR has left the depot and started its pick tour.

- In Chapter 6, we benchmark interventionist and select interventionist approaches

to CHR-OPS against no intervention CHR-OPS and interventionist manual order picking using multiple performance measures like average picking completion time, tardiness, and distance traveled by human picker.

- In Chapter 6, we analyze the impact of multiple factors, including sequencing rules (S-shape vs Largest Gap), the capacity and speed of robots and humans, the number of zones, and heterogeneous order arrival rates by both time and zone.

1.3 Dissertation Outline

The remainder of this dissertation is organized as follows. Chapter 2 introduces the existing literatures related to our work, including traditional OPS, CHR-OPS and dynamic online OPS. Chapter 3 describes the mixed integer linear programming model for the swarm type CHR-OPS using mixed integer linear programming (MILP) with the objective of the total tardiness time. A small set of instances will be tested on the model using GAMS and Gurobi. In Chapter 4, a metaheuristic approach is introduced, called variable neighborhood descent (VND). Such mechanism can better explore and exploit large search space. Sample data will run on MILP programming and variable neighborhood discent to compare the performance. A stochastic metaheuristic approach is introduced in Chapter 5 called the restarted simulated annealing with adaptive neighborhood search (RSA-ANS). In Chapter 6 we introduce an inter-

ventionist approach dealing with dynamic arriving orders. Finally Chapter 7 will be the conclusion and future work plans.

Chapter 2

Literature Review

A considerable amount of prior research has focused on the design and control of picking operations. A comprehensive review of manual and automated OPS is provided by de Koster et al. (2007) and Jaghbeer et al. (2020), respectively. In recent years, human-robot collaboration has been studied in manufacturing domains such as the assembly process (Rahman & Wang, 2015), hazard manufacturing environment (Liu & Wang, 2020), and production workflow (Ferreira et al., 2021). However, research on CHR-OPS is still in its infancy (Azadeh et al., 2020). In this section, we first review a few notable prior research on traditional OPS, as the decisions and solution approach associated with the human-only picking process (e.g., batching, routing) are also applicable to a CHR-OPS.Subsequently, we also discuss the relevant works in CHR-OPS. Then we discuss about the researches about dynamic online

order picking.

2.1 Traditional Human-only Order Picking System

With regards to the traditional human-only OPS, many prior works focus on optimizing one or more sub-problems (order batching, batch assignment and sequencing, and picker routing), while considering other decisions to be fixed or known (Scholz et al., 2017). In particular, the grouping of orders into different batches (i.e., OBP) has been studied extensively in the literature (Henn, 2015). A majority of these works formulated an optimization model to solve small and medium instances (Bahçeci & Öncan, 2021; Gademann & van de Velde, 2005). For instance, Henn (2015) solved the OBP problem to optimality for instances with up to 50 orders. To handle larger instances, several other approaches such as rule-based algorithms, local search, and metaheuristic approaches have been developed. The review article by Çağla Cergibozan & Tasan (2019) summarizes the different solution approaches for the OBP.

In recent years, several research works have simultaneously solved multiple subproblems pertaining to the traditional human-only OPS using different metaheuristic approaches (e.g., Chen et al., 2015; Cheng et al., 2015; Henn, 2015; Scholz et al., 2017). Henn (2015) tackled the order batching and sequencing problem (OBSP) together with multiple pickers, while considering the routing strategy to be fixed based

on a heuristic. They developed variable neighborhood descent (VND) and variable neighborhood search (VNS) algorithms to solve the two subproblems with the goal of minimizing the total tardiness. The authors reported both approaches to yield lower tardiness than priority heuristics. Chen et al. (2015) addressed the order batching, sequencing and routing problem to minimize total tardiness for a warehouse with a single picker. They developed a genetic algorithm (GA) for solving the batching and sequencing problems, and leveraged ant colony optimization (ACO) for determining picker route. Ardjmand et al. (2018) considered the order assignment, batching and multi-picker routing in a wave picking warehouse. They developed VND and hybrid SA algorithms and found the latter to perform slightly better for minimizing makespan. Likewise, Ardjmand et al. (2019) proposed a SA algorithm for the order batching and picker routing problem (OBPRP), and found it to achieve better solution quality when compared to GA. The simultaneous consideration of all the subproblems for a traditional OPS was first addressed by Scholz et al. (2017). The authors developed an optimization model with the objective of minimizing total tardiness. Further, a VND algorithm was developed to handle large instances. They demonstrated that a joint approach to solving the subproblems yielded up to 84% reduction in total tardiness, as opposed to tackling them sequentially. In addition, they demonstrated that metaheuristic approaches, such as VND, achieve substantially better performance than rule-based algorithms.

2.2 Collaborative Human-Robot Order Picking System

One of the first works on human-robot collaboration in a picker-to-parts system was conducted by Lee & Murray (2019). They modeled the collaborative picking operation as a variant of the vehicle routing problem with the objective of minimizing the makespan. Specifically, they focused on understanding the impacts of robot fleet composition and warehouse layout on picking efficiency. A MILP model was developed to address the problem, and extensive numerical analyses were conducted on small test instances. They found that overall performance reduces as the number of cross aisles in the warehouse increases. In addition, the authors also concluded that a centrally positioned depot (or packing station) achieves superior performance over depots located in the warehouse's periphery. Fager et al. (2021) considered a unique picking system, where the human picker collects and places the items in a cart, and a robot that is on board a cart is tasked with sorting the items. They focused on understanding the economic benefits of using such a collaborative system as opposed to an OPS with manual sorting and modeled the relative costs associated with workers, equipment, and quality. They conducted numerical experiments to identify the conditions/settings under which robot-assisted sorting is justifiable.

Recent studies also focused on addressing the traditional decision problems associated with order picking for a CHR-OPS. Azadeh et al. (2020) considered an AMR-assisted picking process and investigated the impact of dynamic switching between

two zoning strategies, namely no zoning and progressive zoning, for a CHR-OPS. They proposed a two-stage solution approach, where a queuing network model is first developed to determine the pick throughput rate, and a Markov decision process model is then employed to assess the picker performance under dynamic switching. Their numerical analysis revealed up to 7% cost savings when the number of zones was dynamically changed. On the other hand, Löffler et al. (2021) considered a picker-lead warehouse system, where robots follow a single picker during the picking process. They addressed the picker routing problem by solving it as a clustered traveling salesman problem with the objective of minimizing the total distance traveled. The routing is done by considering two cases of sequencing the orders, namely, fixed (or given) and open. For the former case, they proposed an extension of the algorithm of Ratliff & Rosenthal (1983), while the latter case is solved using an insertion heuristic. They conducted extensive numerical analysis and showed that the AMR-assisted OPS could reduce the walking distance by up to 20% when benchmarked against a human-based OPS. Recently, Žulj et al. (2021) considered the order batch and batch sequencing problem for an AMR-assisted picker-to-parts system, where the human picker retrieves the item from storage and transfers it at a handover location in the cross-aisle. Therefore, in this system, the picker also uses a cart to carry the item from the pick location to the handover location. They formulate the problem as a mixed integer programming model with the objective of minimizing the total tardiness time. Further, they developed an adaptive large neighborhood search al-

gorithm for the order batching problem (OBP), and a heuristic algorithm for batch sequencing (Žulj et al., 2021). Their numerical analysis showed that the travel speed has a substantial impact on the tardiness as opposed to AMR fleet size.

2.3 Dynamic Online Order Picking System

To achieve a competitive advantage, e-commerce companies offer urgent deliveries, last minute cancellations and allow for frequent and smaller orders. Existing static approaches to OPP are insufficient in today's stochastic setting, where only limited information is available beforehand. Thus, there is a dire need for responsive and flexible methods to handle highly uncertain and rapidly incoming customer orders at warehouses. Recent research has studied various dynamic and interventionist methods to for order picking related activities like:

- **Dynamic warehouse zoning:** In large warehouses, zoning is an approach through which the order-picking area is classified into different zones, and each picker is designated to pick in a particular zone (Koster et al., 2007). In an R-OPP system, one or more pickers are restricted to each zone, while the AMR can visit any zone. Zoning thus can limit the assigned picker's walking distances and improve efficiency. On the contrary, due to demand variability, zoning may also result in additional waiting times for AMRs and compromise performance

(Azadeh et al., 2020). Azadeh et al. (2020) studied the R-OPP with no zoning and progressive zoning and reported that the average order size determines the optimal zoning policy. They propose a Markov Decision Process (MDP) model to enable a dynamic zoning policy based on the number of small and large sized customer orders in the warehouse system.

- **Dynamic order batching:** Batching is the process of grouping multiple orders so they can be picked in a single pick cycle to minimize the average pick time per order (Giannikas et al., 2017; Yu and De Koster, 2019). Depending on when the order information is available, batching can be static/ offline or dynamic/ online. In the former, the customer orders are known at the start of the planning horizon (e.g., shift, day). In contrast, for dynamic/ online batching, the stochastic property of the incoming customer orders needs to be considered, and the information on actual orders only becomes available dynamically over time (Giannikas et al., 2017; Yu and De Koster, 2019). Giannikas et al. (2017) indicate that the common methods for dynamic order batching are (i) Fixed Time Window Batching (FTWB) and (ii) Variable Time Window Batching (VTWB). In FTWB, all incoming orders within a fixed time window are grouped into a single batch (Van Nieuwenhuyse and De Koster, 2009; Henn et al., 2012; Schleyer and Gue, 2012; Zhang et al., 2017). In VTWB all incoming orders until a fixed threshold quantity (Van Nieuwenhuyse and De Koster, 2009; Xu et al., 2014), or some other criteria (Bukchin et al., 2012; Pérez-Rodríguez et al., 2015) is

reached are grouped together. Finally, hybrids of FTWB and VTWB have also been proposed (Zhang et al., 2016).

- **Dynamic pick lists:** Most studies focus only on dynamic order batching where in once the picker starts the pick-cycle, the pick list can't be modified. However, such an approach may compromise on opportunities to fulfill some urgent orders or effectively pick new orders that are along the picking tour but were received after the onset of the tour (Gong and De Koster, 2008; Rubrico et al., 2011; Giannikas et al., 2017). To this end, Rubrico et al. (2011) and Gong and De Koster (2008) have proposed approaches/ systems where the current pick list can be modified during the pick cycle to include new orders that can be picked downstream along with the remaining un-picked orders.

- **Dynamic picker routing:** Once a pick list is dynamically updated while the picker is on their route, there is a need to intervene and update the route of the picker. To this end, Lu et al., (2016) propose an Interventionist Routing Algorithm (IRA) that considers the pickers current location, remaining picking locations, and the depot location (i.e., destination) and re-routes the picker every time the pick list is updated. They report that the IRA approach to dynamic OPP routing outperforms both optimal static and dynamic heuristic methods.

2.4 Research Gaps

2.4.1 Research Gaps of CHR-OPS with Static Orders

In conclusion, our review of relevant literature provided the following insights on OPS. First, very limited research has considered all the subproblems together for the traditional OPS. In the case of CHR-OPS, prior research that solves them simultaneously is almost non-existent. Second, studies on collaborative order picking do not focus on minimizing tardiness, though Žulj et al. (2021) is a notable exception. However, it is crucial to reduce tardiness in practice as delayed shipments result in fines, higher supply chain costs, and customer dissatisfaction (Scholz & Wäscher, 2017). The other objectives, such as minimizing makespan or travel distance, do not consider the order due date, and therefore may not be suitable for mitigating delayed shipments. Third, metaheuristic approaches, such as VND and VNS, are well-suited to achieve good solution quality as opposed to priority rule-based heuristics. However, existing works have not employed such an approach for solving a swarm-type CHR-OBASRP. Fourth, despite some similarities between traditional and collaborative OPS, there exist many differences - (i) unlike traditional OPS, each item must be assigned to two resource types (picker and AMR) in the case of CHR-OPS, (ii) a routing plan that is coordinated among the two resource types (AMRs and pickers) is needed for a CHR-OPS, but traditional human-only OPS involves route planning

for a single resource type (i.e., pickers), (iii) the picker is not required to push a cart or return to the packing station in a CHR-OPS, and hence the order batching as well as batch sequencing decisions do not apply to the picker. Thus, it is necessary to develop solution approaches that consider the unique characteristics of a CHR-OPS. This research (i.e. Chapters 3, 4, and 5) contributes to the literature on static OPS by developing optimization models and efficient solution approaches for the CHR-OBASRP. Table 2.1 summarizes the relevant works on static OPS and highlights the contributions of this research.

Table 2.1: Summary of relevant works on traditional and collaborative order picking

Reference	**Objective**	**Decisions**				**OPS Characteristics**			
		Batching	**Assignment**	**Sequencing**	**Routing**	**No. of Pickers**	**No. of Robots**	**SKUs per Order**	**Order Splitting**
Traditional Human-Only OPS									
Henn (2015)	tardiness	✓		✓		M	NA	M	N
Cheng et al. (2015)	distance	✓			✓	1	NA	1	NA
Scholz et al. (2017)	tardiness	✓	✓	✓	✓	M	NA	M	N
Li et al. (2017)	distance	✓			✓	1	NA	1	NA
Valle et al. (2017)	distance	✓			✓	M	NA	1	NA
Ardjmand et al.(2018)	makespan	✓	✓	✓	✓	M	NA	M	N
Kuhn et al. (2020)	tardiness	✓	✓	✓	✓	M	NA	M	N
Human-Robot OPS									
Lee & Murray(2019)	makespan			✓	✓	M	M	1	NA
Azadeh et al. (2020)	makespan			✓	✓	M	M	M	N
Fager et al.(2021)	cost	✓				1	1	M	N
Löffler et al. (2021)	distance	✓		✓	✓	1	M	M	N
Zulj et al. (2021)	tardiness	✓		✓	✓	M	M	1	NA
Proposed Research(Chapter 3-5)	**tardiness**	✓	✓	✓	✓	**M**	**M**	**M**	**Y**

M: Multiple; NA: Not Applicable; Y: Allowed; N: Not Allowed

2.4.2 Research Gaps of CHR-OPS with Dynamic Orders

The following is the overview of the limitations in the existing literature pertaining to HRC-OPP with dynamic orders.

- Most of the extant literature focuses on the static order picking problem (for both traditional and collaborative settings). These approaches cannot be effectively scaled for dynamic systems, which is critical for most warehouses dealing with tight due dates and B2C orders.
- Although dynamic appraoches for traditional human-only picking system have received some attention in the literature (Giannikas et al., 2017; Yu and De Koster, 2019; Rubrico et al., 2011; Gong and De Koster, 2008), there is very limited work focusing on dynamic R-OPP.
- While few R-OPPs consider dynamic order batching, they do not allow for interventions that modify the pick list after the robot leaves the depot, limiting the system's responsiveness. The inability to modify the pick list dynamically to pick new orders downstream of the pick tour will compromise system performance.
- Today's warehouses are large and proceess high volumes of orders, and hence will benefit from collaboration of multiple pickers and robots. Nevertheless, research on multiple pickers and robots in a dynamic environment is limited.
- Finally, collaborative human-robot order-picking systems are in their early stages.

To our best knowledge, there are no studies that benchmark interventionist approach to CHR-OPP against traditional systems using performance measures relevant to warehouse managers.

Table 2.2: Summary of relevant works on dynamic order picking

Reference	**Objective**	**Decisions**				**Collaborative OPS**	**OPS Characteristics**			
		Batching	**Assignment**	**Sequencing**	**Routing**		**No. of Pickers**	**No. of Robots**	**SKUs per Order**	**Order Splitting**
Gong & Koster (2008)	cost					N	M	NA	1	N
Wenrong et al. (2016)	distance				✓	N	1	NA	1	N
Giannikas et al. (2017)	time	✓	✓	✓	✓	N	1	NA	1	N
Zhang et al. (2016)	time	✓	✓	✓		N	M	NA	1	N
Xu et al. (2014)	time	✓				N	M	NA	1	N
Azadeh et al. (2020)	makespan	✓		✓	✓	Y	M	M	1	N
Proposed Research(Chapter 6)	multiple	✓	✓	✓	✓	Y	M	M	1	N

M: Multiple; NA: Not Applicable; Y: Allowed; N: Not Allowed

The existing literature on Robot-assisted Order Picking Problems (R-OPP) predominantly focuses on static demand, while today's warehouses face dynamic orders and uncertainty. Few R-OPPs allow interventions to modify pick lists after the robot departs, impacting system responsiveness. Fixed assignments in Picker-in and AMR-in lead systems limit flexibility compared to swarm systems with synchronized collaboration. Collaborative human-robot order-picking systems are still emerging, warranting benchmarking against manual systems using relevant performance measures. Table 2.2 shows the relevant works about dynamic online order picking. We will address all these gaps mentioned.

This study seeks to address all these points by proposing a CHR-OPS with multiple AMRs and human pickers that collaborate to complete the picking tasks. We also assume that the customer orders will appear dynamically and propose interventionist and select-interventionist algorithms to service them. These algorithms also effectively modify the pick list, allowing for picking new orders after the start of pick cycle. Finally, we also benchmark the proposed interventionist and rule-based interventionist CHR-OPS against non-interventionist CHR-OPS and interventionist manual picking process. Benchmarking is done by evaluating varied performance measures including human travel distance, order tardiness, and order completion time. Finally, we also analyze impact of multiple factors, including sequencing rules (S-shape vs Largest Gap), the capacity and speed of robots and humans, the number of zones, and heterogeneous order arrival rates by both time and zone, to derive managerial insights.

Chapter 3

Optimization Model for Collaborative Order Picking: Integrated Batching, Sequencing and Routing

3.1 Problem Description

In this chapter, we consider a set of orders $j \in \mathcal{J}$, containing $i \in \mathcal{N}$ (where $|\mathcal{N}| \geq |\mathcal{J}|$) numbered items, to be picked from warehouse storage and delivered at a drop-

off point (depot). Specifically, each order $j \in \mathcal{J}$ contains a non-empty set of items $i \in \mathcal{O}_j$, which may correspond to one or more SKUs of varying quantities. The items are stored in a block warehouse with open-end picking aisles, as shown in Figure 3.1. The block layout constitutes picking and cross aisles. The aisles that contain storage racks on both sides and are parallel to each other are referred to as a picking aisle. On the other hand, the pathway that does not contain any storage bins but allows the pickers and transporters to enter/exit the picking aisle is called the cross-aisle. Each rectangular slot represents a storage location and can store multiple items of the same SKU. Moreover, these items are stored in low-level bins, thereby allowing direct access to the picker. Table 3.1 provides an illustration of the information associated with the orders along with the item numbering scheme adopted in this research. Specifically, the items in $|\mathcal{J}|$ orders are numbered consecutively (i.e., $1, 2, ..., |\mathcal{N}|$), while SKU denotes a specific item. Each SKU is stored in a unique location in the warehouse and is specified by the aisle, rack, shelf and position numbers. For instance, A01-R02-S03-P05 denotes the first aisle, second rack, third shelf, and fifth position on that level from left. Thus, if two orders include the same item, then the item SKU and location will be the same but the item numbering will be unique for modeling purposes (e.g., SKU 100001 is present in orders 1 and 2 in Table 3.1). Depending on the order quantity of item numbered $i \in \mathcal{N}$, the number of storage bins (β_i) required by the AMR will be determined. However, two different item numbers $i, i' \in \mathcal{N}$ cannot be placed in the same bin.

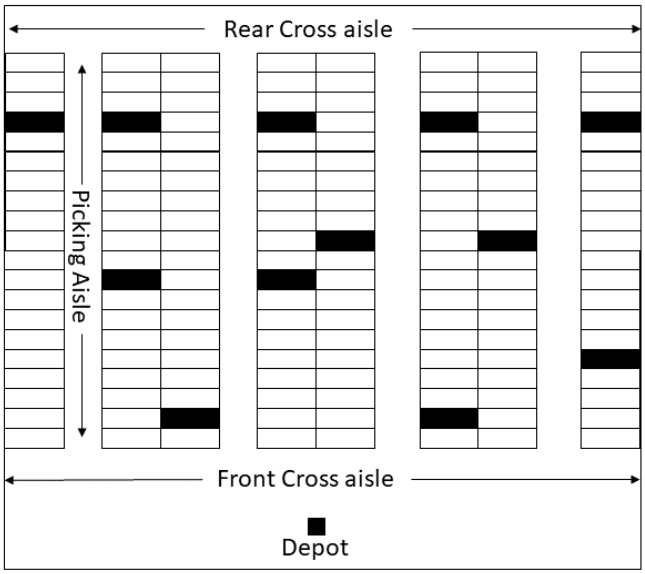


Figure 3.1: Illustration of Block Warehouse Layout

The order picking process involves a fleet of robot transporters ($r \in \mathcal{R}$) and human pickers ($k \in \mathcal{K}$) working in a collaborative manner. The AMRs are responsible for transporting the items to the drop-off point and carry a cart that can hold up to κ_r items. An AMR may make $t \in \mathcal{T}$ tours and transport a subset of $|\mathcal{N}|$ items (i.e., batch) in each tour. The human workers must perform the pick-and-place operation and takes η_i^U time units to retrieve order $i \in \mathcal{N}$ from the rack, and η_i^L time units to load/place that order on the robot cart. The time taken by the AMR and human worker to travel from the pick location of item i' to i is given by $\tau_{i'i}^R$ and $\tau_{i'i}^K$, respectively. Likewise, the tuples $< \tau_{0i}^R, \tau_{i0}^R >$ and $< \tau_{0i}^K, \tau_{i0}^K >$ denote the travel time between location i and depot for the robot and human workers, respectively. The travel times are assumed to be affected by the resource's (AMRs and human pickers) travel speed and warehouse layout.

The picking operation is driven by the worker pick list and AMR mission list. The

Table 3.1: Illustrative example of orders and associated item numbering adopted in optimization model

Order ($j \in \mathcal{J}$)	**SKU**	**Item Numbering** ($i \in \mathcal{N}$)	**Number of Bins Required** (β_i)	**Item Location**
1	100001	1	2	A01-R01-S02-P05
	230653	2	1	A04-R02-S01-P08
	242411	3	2	A04-R04-S01-P03
2	629360	4	1	A14-R02-S03-P10
	100001	5	1	A01-R01-S02-P05
⋮	⋮	⋮	⋮	⋮
$\lvert\mathcal{J}\rvert$	365312	$\lvert\mathcal{N}\rvert - 1$	1	A06-R05-S02-P08
	242411	$\lvert\mathcal{N}\rvert$	2	A04-R04-S01-P03

pick list specifies the set of items, their location, and sequence in which they must be visited by the human workers to perform the pick-and-place operation. Likewise, the mission list provides the AMRs with the number of tours, list of items to be collected in each tour along with their location and sequence. The pick and mission lists typically include items corresponding to different orders, but the same picker or AMR need not be responsible for collecting all the items belonging to an order. A batch is the set of items to be collected by the AMR in a tour, and directly impacts the completion time of order j (C_j). The batch start time is the time at which the AMR begins tour $t \in \mathcal{T}$ from the depot to collect the items in that batch, while the batch completion time refers to the time at which the AMR returns to the depot to drop-off all the items in that batch. An order is complete only if all the batches containing the items pertaining to that order are returned to the depot. Figures 3.2 and 3.3 show the operations sequence from the picker and AMR perspective in a

CHR-OPS, respectively.

Typical of distribution warehouses, each order $j \in \mathcal{J}$ has a due date d_j to ensure that the truck carrying this order can leave the facility as planned and deliver it to the external customer within the expected time window (Lee & Murray, 2019; Žulj et al., 2021). As illustrated by Scholz et al. (2017), the extent to which a due date is met depends on how well the key subproblems are solved. In other words, the items in a batch, assignment of batches to AMR, order in which the batches are processed, and the routing of AMRs, as well as pickers, affect the completion time of the order. A delayed fulfillment of order results in penalties (in the form of customer dissatisfaction or lost revenue) and is typically dependent on the delay duration (tardiness). Specifically, the tardiness associated with an order is the non-negative difference between order completion time and due date (i.e., $T_j = \max\{C_j^J - d_j, 0\}$). The total tardiness, which is the sum of tardiness associated with all orders, is an important metric pertaining to fulfillment efficiency in the literature on order picking (Žulj et al., 2021) as well as other problem areas involving customer deadlines. Given the aforementioned characteristics, the CHR-OBASRP seeks to minimize the total tardiness by optimizing the following three subproblems associated with collaborative order picking.

(i) How to partition the $|\mathcal{N}|$ items to a set of batches? How many items should be included in a batch?

(ii) How to assign the batches to the set of AMRs, and in which sequence (or tour) should an AMR process the assigned batches?

(iii) Given a set of items to be collected in a batch, in what order should their locations be visited by the AMR. Likewise, how should the human worker be routed to coordinate the picking process?

To model the CHR-OBASRP, we have made the following assumptions. First, we consider all the pickers to walk at the same speed and all the robots to move at a constant speed. Second, the AMRs and pickers can travel in both directions in an aisle, and they always take the shortest path when traveling from one location to another. Third, picking and cross aisles are assumed to be wide enough for robots and humans to overtake or pass through from opposite directions. Therefore, the possibility of a robot or picker waiting behind another resource (blocking) is not considered. Finally, the robot and picker start (and end) the pick operation at the depot. Note that the picker need not return to the depot at the end of each robot tour but is assumed to return once all the items have been collected.

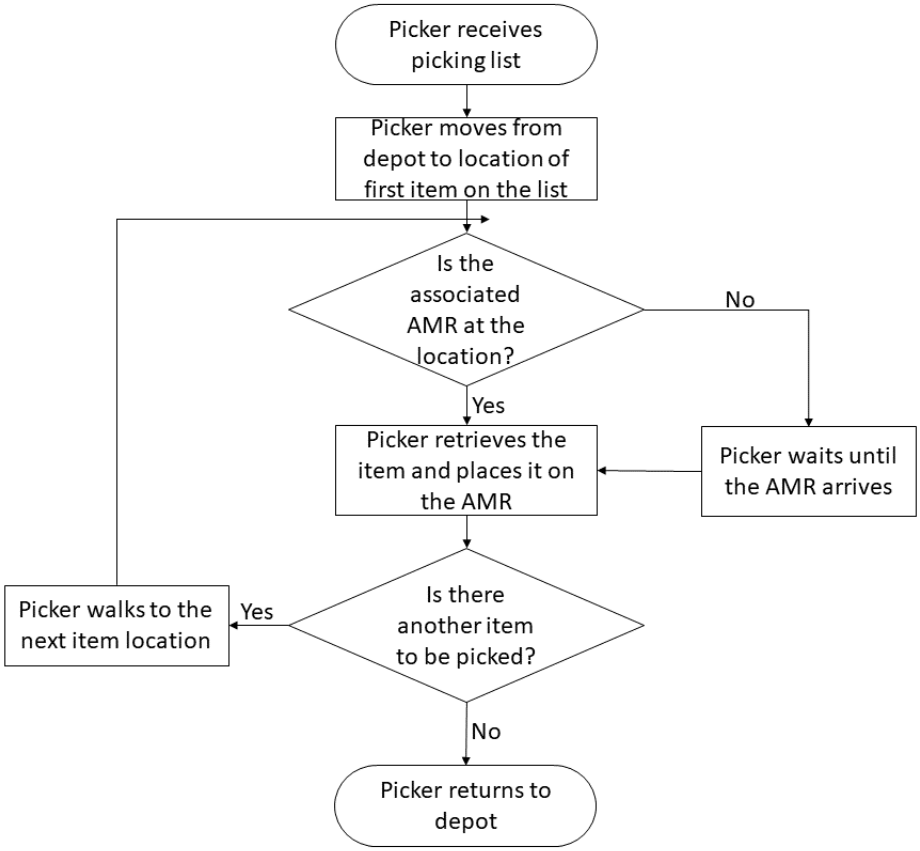


Figure 3.2: Pick-and-place process from human picker's perspective in a CHR-OPS

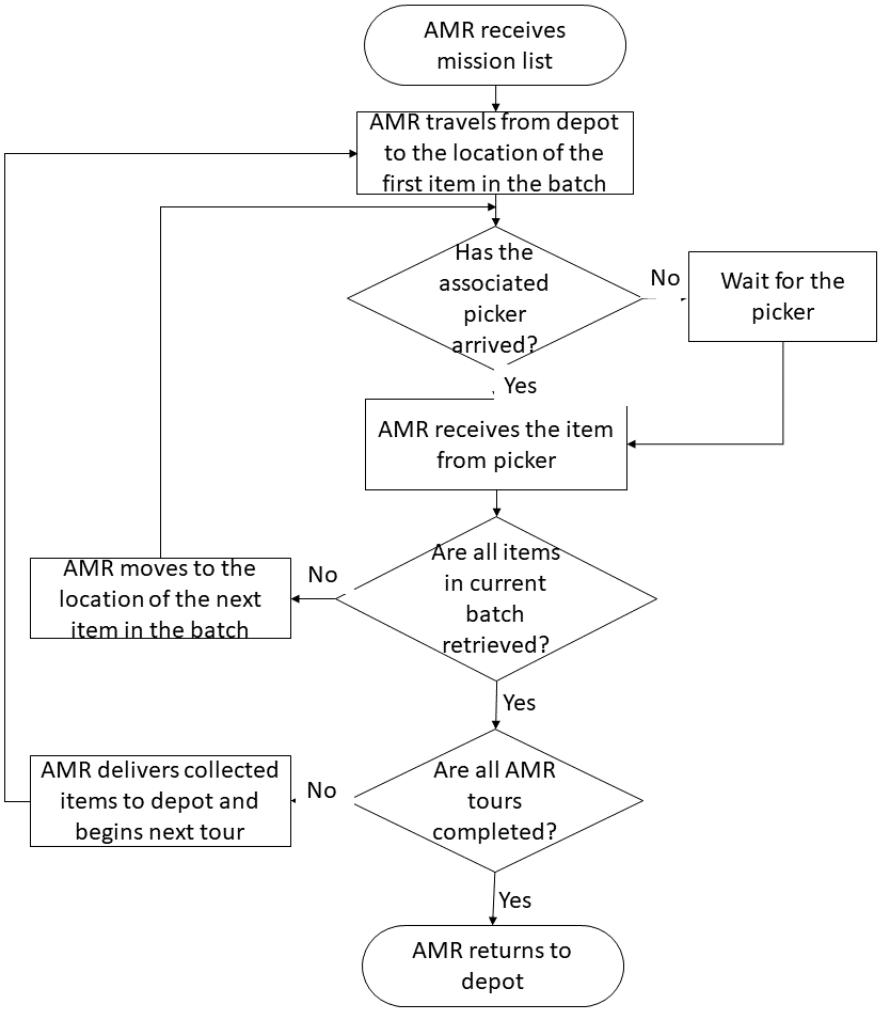


Figure 3.3: Item collection process from AMR's perspective in a CHR-OPS

3.2 Notations

Indices and Sets

$j \in \mathcal{J}$ Set of orders

$i \in \mathcal{N}$ Set of consecutive numbering of items in $|\mathcal{J}|$ orders

$i \in \mathcal{O}_j$ Set of items corresponding to order j

$r \in \mathcal{R}$	Set of autonomous mobile robots (AMRs), $\mathcal{R} = \{r_1, r_2, ..., r_{\max}\}$
$k \in \mathcal{K}$	Set of human pickers, $\mathcal{K} = \{k_1, k_2, ..., k_{\max}\}$
$t \in \mathcal{T}$	Set of maximum allowable AMR tours (or trips)

Parameters

M	Large positive number
$\tau^K_{i',i}$	Travel time from location of item i' to location of item i by human pickers
$\tau^R_{i',i}$	Travel time from location of item i' to location of item i by AMR
κ_r	Capacity of robot r in any tour, expressed as the number of bins which can be carried by the AMR
β_i	Number of bins required to carry item number i from storage to drop-off point
η^U_i	Time to retrieve (unload) item i from storage location
η^L_i	Time to place (load) item i on AMR's cart
d_j	Due date for order j

Decision Variables

$A^K_{i,k}$	1 if operator k is assigned to pick-and-place item i; 0 otherwise
$A^R_{i,r,t}$	1 if item i is batched to be transported by robot r in tour t; 0 otherwise
$X^R_{r,t}$	1 if robot r makes a non-empty tour t

$U^K_{i',i}$	1 if operator k retrieves item i' before item i, but not necessarily exactly before item i (i.e., relative precedence or indirect sequencing); 0 otherwise
$U^R_{i',i}$	1 if robot r collects item i' before item i in the same trip but not necessarily exactly before item i (i.e., relative precedence or indirect sequencing); 0 otherwise
$B^K_{i,k}$	Retrieval begin time of item i by picker k
$F^K_{i,k}$	Retrieval finish time of item i by picker k
$L^K_{i,k}$	Time at which picker k is ready to leave location of item i
$F^R_{i,r,t}$	Collection finish time of item i by AMR r in tour t
$B^R_{i,r,t}$	Collection begin time of item i by AMR r in tour t
C^K_k	Return-to-base time of worker k after completing the assigned pick-and-place tasks
$C^R_{r,t}$	completion time of tour t by robot r
S^K_k	Start time of operator k from depot (base) to perform pick-and-place task
$S^R_{r,t}$	Start time of tour t by robot r
C_j	Delivery completion time of all items pertaining to order j
T_j	Tardiness for order j

3.3 CHR-OBASRP Model Formulation

$$\min \quad Z = \sum_{j \in \mathcal{J}} T_j \tag{3.1}$$

subject to

$$\sum_{k \in \mathcal{K}} A^K_{i,k} = 1 \qquad \forall i \in \mathcal{N} \tag{3.2}$$

$$\sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}^{\mathcal{R}}} A^R_{i,r,t} = 1 \qquad \forall i \in \mathcal{N} \tag{3.3}$$

$$X^R_{r,t} \leq \sum_i A^R_{i,r,t} \qquad r \in \mathcal{R}, \forall t \in \mathcal{T} \tag{3.4}$$

$$\sum_i \beta_i \cdot A^R_{i,r,t} \leq \kappa_r \cdot X^R_{r,t} \qquad r \in \mathcal{R}, \forall t \in \mathcal{T} \tag{3.5}$$

$$X^R_{r,t} \geq X^R_{r,t+1} \qquad r \in \mathcal{R}, t \in \mathcal{T}, t < |\mathcal{T}| \tag{3.6}$$

$$A^K_{i,k} - A^K_{i',k} \leq 1 - (U^K_{i,i'} + U^K_{i',i}) \qquad i, i' \in \mathcal{N}, i \neq i', k \in \mathcal{K} \tag{3.7}$$

$$A^K_{i,k} + A^K_{i',k} \leq 1 + (U^K_{i,i'} + U^K_{i',i}) \qquad i, i' \in \mathcal{N}, i \leq i', k \in \mathcal{K} \tag{3.8}$$

$$A^R_{i,r,t} - A^R_{i',r,t} \leq 1 - (U^R_{i,i'} + U^R_{i',i}) \qquad i, i' \in \mathcal{N}, i \neq i', r \in \mathcal{R}, t \in \mathcal{T} \tag{3.9}$$

$$A^R_{i,r,t} + A^R_{i',r,t} \leq 1 + (U^R_{i,i'} + U^R_{i',i}) \qquad i, i' \in \mathcal{N}, i \leq i', r \in \mathcal{R}, t \in \mathcal{T} \tag{3.10}$$

$$B^K_{i,k} \geq S^K_k + \tau^K_{0i} - M \cdot (1 - A^K_{i,k}) \qquad i \in \mathcal{N}, k \in \mathcal{K} \qquad (3.11)$$

$$\sum_{k \in \mathcal{K}} B^K_{i,k} \geq \sum_{k \in \mathcal{K}} L^K_{i',k} + \tau^K_{i',i} \cdot U^K_{i',i} - M \cdot (1 - U^K_{i',i}) \qquad i, i' \in \mathcal{N}, k \in \mathcal{K}, i \neq i' \qquad (3.12)$$

$$F^K_{i,k} = B^K_{i,k} + \eta^U_{i,k} \cdot A^K_{i,k} \qquad i \in \mathcal{N}, k \in \mathcal{K} \qquad (3.13)$$

$$L^K_{i,k} \geq F^K_{i,k} \qquad i \in \mathcal{N}, k \in \mathcal{K} \qquad (3.14)$$

$$L^K_{i,k} \geq \sum_{t \in \mathcal{T}} F^R_{i,r,t} - M \cdot (2 - A^K_{i,k} - \sum_{t \in \mathcal{T}} A^R_{i,r,t}) \qquad i \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R} \qquad (3.15)$$

$$B^K_{i,k} \leq M \cdot A^K_{i,k} \qquad i \in \mathcal{N}, k \in \mathcal{K} \qquad (3.16)$$

$$F^K_{i,k} \leq M \cdot A^K_{i,k} \qquad i \in \mathcal{N}, k \in \mathcal{K} \qquad (3.17)$$

$$L^K_{i,k} \leq M \cdot A^K_{i,k} \qquad i \in \mathcal{N}, k \in \mathcal{K} \qquad (3.18)$$

$$B^R_{i,r,t} \geq S^R_{r,t} + \tau^R_{0i} - M \cdot (1 - A^R_{i,r,t}) \qquad i \in \mathcal{N}, r \in \mathcal{R}, t \in \mathcal{T} \qquad (3.19)$$

$$\sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}^{\mathcal{R}}} B^R_{i,r,t} \geq \sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}^{\mathcal{R}}} F^R_{i',r,t} + \tau^R_{i',i} \cdot U^R_{i',i} - M \cdot (1 - U^R_{i',i}) \qquad i, i' \in \mathcal{N}, r \in \mathcal{R}, i \neq i' \qquad (3.20)$$

$$\sum_{t\in\mathcal{T}} B^R_{i,r,t} \geq \sum_{k\in\mathcal{K}} F^K_{i,k} - M\cdot(2-\sum_{k\in\mathcal{K}} A^K_{i,k} - \sum_{t\in\mathcal{T}} A^R_{i,r,t}) \qquad i\in\mathcal{N}, k\in\mathcal{K}, r\in\mathcal{R} \qquad (3.21)$$

$$F^R_{i,r,t} = B^R_{i,r,t} + \eta^L_{i,r}\cdot A^R_{i,r,t} \qquad \forall i\in\mathcal{N}, r\in\mathcal{R}, t\in\mathcal{T} \qquad (3.22)$$

$$B^R_{i,r,t} \leq M\cdot A^R_{i,r,t} \qquad i\in\mathcal{N}, r\in\mathcal{R}, t\in\mathcal{T} \qquad (3.23)$$

$$F^R_{i,r,t} \leq M\cdot A^R_{i,r,t} \qquad i\in\mathcal{N}, r\in\mathcal{R}, t\in\mathcal{T} \qquad (3.24)$$

$$C^R_{r,t} \geq F^R_{i,r,t} + \tau^R_{i0} - M\cdot(1-A^R_{i,r,t}) \qquad i\in\mathcal{N}, r\in\mathcal{R}, t\in\mathcal{T} \qquad (3.25)$$

$$S^R_{r,t} \geq C^R_{r,t-1} - M\cdot(1-X^R_{r,t}) \qquad t\in\mathcal{T}, r\in\mathcal{R} \qquad (3.26)$$

$$C_j \geq C^R_{r,t} - M\cdot(1-A^R_{i,r,t}) \qquad \forall i\in\mathcal{O}_j, t\in\mathcal{T}, r\in\mathcal{R}, j\in\mathcal{J} \qquad (3.27)$$

$$T_j \geq C_j - d_j \qquad \forall j\in\mathcal{J} \qquad (3.28)$$

$$S^R_{r,t} \leq M \cdot X^R_{r,t} \qquad r \in \mathcal{R}, t \in \mathcal{T} \tag{3.29}$$

$$C^R_{r,t} \leq M \cdot X^R_{r,t} \qquad r \in \mathcal{R}, t \in \mathcal{T} \tag{3.30}$$

$$A^K_{i,k}, A^R_{i,r,t}, X^R_{r,t}, U^K_{i',i}, U^R_{i',i} \in \{0,1\} \qquad i, i' \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R}, t \in \mathcal{T} \tag{3.31}$$

$$B^K_{i,k}, F^K_{i,k}, L^K_{i,k}, F^R_{i,r,t}, B^R_{i,r,t}, C^K_k, C^R_{r,t}, S^K_k, S^R_{r,t}, C_j, T_j \geq 0 \quad i \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R}, t \in \mathcal{T}, j \in \mathcal{J} \tag{3.32}$$

The MILP model for CHR-OBASRP is given by Equations (3.1) - (3.32). The objective function (Equation 3.1) seeks to minimize the total tardiness. Constraint (3.2) ensures that the pick location of every item to be retrieved is assigned to be visited by exactly one operator, while constraint (3.3) guarantees that each item to be transported is assigned exactly to one AMR tour. Although the AMRs are allowed to make up to $|\mathcal{T}|$ tours, it may not need to make all the allowable tours to transport the $|\mathcal{N}|$ items. Therefore, constraints (3.4) - (3.5) determine if AMR r makes a non-empty tour t. If no items are assigned to AMR r in tour t (i.e., $\sum_i A^R_{i,r,t} = 0$), then constraint (3.4) will ensure that tour t is not made by that AMR (also referred to as an *empty tour* in this paper) by forcing $X^R_{r,t}$ to zero. However, if AMR r is assigned to transport at least one item in tour t (i.e., $\sum_i A^R_{i,r,t} \geq 1$), then constraint

(3.5) will force that AMR to make a non-empty tour t (i.e., $X^R_{r,t} = 1$). In addition, constraint (3.5) also ensures that the total number of bins required to transport the items collected in an AMR tour does not exceed the AMR capacity κ_r. Since an AMR is expected to make successive non-empty tours until all items assigned to that AMR are delivered, it is not practical to have an empty tour between two non-empty tours. Thus, constraint (3.6) is introduced to prevent the AMR from performing a non-empty tour $t + 1$ if that AMR was not assigned to transport any items in the previous tour t. So, when tour t for AMR r is an empty tour (i.e., $X^R_{r,t} = 0$), then constraint (3.6) forces the next tour $t + 1$ for that AMR to be an empty tour as well. However, if AMR r makes a non-empty tour t, then constraint (3.6) allows the next tour $t + 1$ to either be an empty or non-empty tour. The relative precedence (or indirect sequencing) between two items, i and i', assigned to the same operator is ensured using Constraints (3.7) - (3.8). Therefore, if items i and i' are assigned to the same operator k, then Constraints (3.7) - (3.8) force either $U^K_{i,i'}$ or $U^K_{i',i}$ to be one. However, when they are not allocated to the same operator, then both $U^K_{i,i'}$ or $U^K_{i',i}$ will be zero. Likewise, if items i and i' are transported in the same AMR tour, then their relative precedence (or indirect sequencing) is guaranteed by Constraints (3.9) - (3.10).

The retrieval begin time of an item from storage is controlled by constraints (3.11) and (3.12). Specifically, the time at which the picker can start retrieving the item is the maximum of two events - (i) travel time from base to the pick location or (ii) time

at which the operator can depart the previous pick location plus the travel time to the current pick location. The finish time of the pick operation is the sum of retrieval start time and unload time (includes search and pick time), as given by constraint (3.13). The earliest time at which an operator can leave a pick location is given by constraints (3.14) and (3.15). If an item is not assigned to a picker, then he/she cannot perform the pick-and-place operation, and therefore the corresponding tasks are forced to zero by constraints (3.16) - (3.18).

Likewise, the start time of placing item i on an AMR is governed by constraints (3.19) - (3.21). Specifically, the AMR cannot collect the item before it arrives to the pick location of that item, as given by constraints (3.19) and (3.20). In addition, the AMR can collect the item only after the human worker completes the pick operation (constraint 3.21). The collection finish time of an item by an AMR is the sum of collection begin time and loading time, and is determined by constraint (3.22). However, if an item is not assigned to be transported by an AMR in a specific tour, then the related variables are forced to zero by constraints (3.23) - (3.24).

The AMR tour completion time (equivalently batch completion time) is determined by constraint (3.25). On the other hand, constraint (3.26) ensures that an AMR can begin a tour only after the completion of the previous tour. Since the AMRs are assumed to be ready at the beginning of the picking process, $C^R_{r,0}$ is set to zero in constraint (3.26) for all AMRs. The delivery completion time of all the items in an

order is given by constraint (3.27), and the corresponding tardiness is determined by constraint (3.28). Nevertheless, if the AMR does not make a tour, then the start and completion times of the associated tour are restricted to zero by constraint (3.30). Finally, the non-negativity and binary restrictions on the decision variables are specified by constraints (3.31) - (3.32), respectively.

3.4 Solution Representation

The complete solution to the CHR-OBASRP includes two parts - mission list of all AMRs and pick list of all workers. The mission list specifies the items to be picked by AMR $r \in \mathcal{R}$ in tour $t \in \mathcal{T}$ along with their sequence. On the other hand, the pick list provides the sequence of pick locations to be visited by human worker $k \in \mathcal{K}$. Figure 3.4 provides a general illustration of the solution to CHR-OBASRP. For example, the third item to be collected by a robot or picker in a trip is denoted as $i_{[3]}$, and the last item in a trip is represented as $i_{[\text{end}]}$. Therefore, $i_{[3]}$ corresponds to different items in each tour made by the operator and AMR. The total number of items carried in a tour by robot $r \in \mathcal{R}$ cannot exceed its cart capacity κ_r.

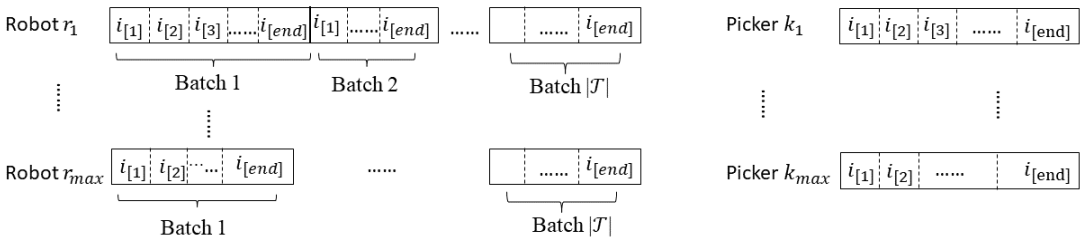


Figure 3.4: General Illustration Solution of CHR-OBASRP

Fig 3.5 shows the solution representation for a numerical example, where 15 items are being picked by 2 human workers and transported by 2 AMRs who can each make up to 3 tours. As shown in Figure 3.5, the first robot will collect items 5, 9 and 8 sequentially (i.e., $5 \rightarrow 9 \rightarrow 8$) from their pick locations, and return to the depot to complete the first tour. Also, Robot 1 makes all three allowable tours, but Robot 2 makes only two tours. Since no items are assigned in the third batch for Robot 2, it is represented as an empty tour (i.e., the AMR does not make the third trip). Likewise, Picker 1 will pick-and-place item 5 first and then proceed to the location of item 9, and the remaining item locations in the specified sequence. The picker does not return to the depot for each batch of items collected by the robot, and therefore the representation of the pick lists does not contain multiple batches.

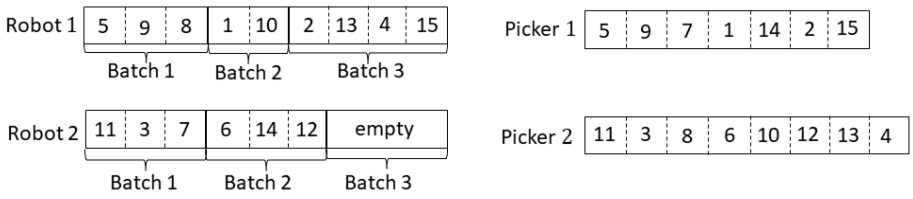


Figure 3.5: Numerical illustration of mission and pick lists

3.5 Generation of Problem Instances

To generate realistic test instances, we set the warehouse attributes based on the prior literature (Lee & Murray, 2019). Specifically, a single-block warehouse with 20 storage racks is considered, where each rack is 20 feet long and each storage location is 1-foot wide by 5 feet deep (Lee & Murray, 2019). The racks are arranged in parallel such that the picking area is divided into 10 five-foot-wide picking aisles and 2 horizontal cross-aisles (Pan et al., 2014). Each picking aisle provides access to 40 storage locations (i.e., each side of the aisle has 20 racks), thereby resulting in a total of 400 storage locations. In addition, a single depot (or drop-off station) is located in the middle of the front horizontal cross-aisle.

To create a test instance, we randomly generate $|\mathcal{N}|$ item locations based on the 400 storage positions. For generating small test instances, we consider a total of 10 or 15 items split among 5 and 7 orders, respectively. Also, our experimentation explores four different levels of human-robot team compositions $(p_{\max}, r_{\max})$ for collaborative order picking - (1,1), (2,1), (1,2) and (2,2) for small instances. For the baseline setting, we set the AMR cart capacity and travel speed to 20 items and 2 feet/second, respectively. On the other hand, human pickers walk at a speed of 1 foot/second (Lee & Murray, 2019). These speeds and warehouse configuration are used to establish the travel time of the AMRs and workers within the warehouse (e.g., depot to a storage location, storage location of one item to another). The item retrieval time for

a human worker is 0.75 seconds, and the time taken to place it on the AMR is also 0.75 seconds (Lee & Murray, 2019).

To generate the due dates, we adapt the procedure followed by previous work on traditional human-only OPS (Henn, 2015; Scholz et al., 2017). Specifically, the generation of due dates depends on four factors - pick completion time of order j ($\hat{C}_j$) when it is the only order processed in a tour, available AMRs ($r_{\max}$) and pickers ($p_{\max}$), and the modified traffic congestion rate (γ), a parameter that affects due date tightness (Lee & Murray, 2019). Given these four factors, the due date associated with order $j \in \mathcal{J}$ is randomly generated from the interval $\big[\hat{C}_j, (2 \cdot (1-\gamma) \sum_{j \in \mathcal{J}} \hat{C}_j + \min_{j \in \mathcal{J}} \hat{C}_j)/\min\{p_{\max}, r_{\max}\}\big]$. Consistent with prior research, we fix the value of γ as 0.6, 0.7 or 0.8 (Henn, 2015; Scholz et al., 2017).

In this dissertation, a problem class is established based on three parameters, namely, $\mathcal{N}$, $(p_{\max}, r_{\max})$, and γ. Based on the above described values of these three parameters, we generate 12 different problem classes.

3.6 Result and Conclusions

In this section, we solve the proposed MILP model and verify the solution obtained. To assess the accuracy and correctness of the MILP model, we manually compute performance measures based on the optimal value of the decision variables

and compare them against the solution obtained. We found the two solutions (manually computed vs. model output) to be in complete agreement for the small instances that are randomly selected. Furthermore, we also evaluated the model for certain edge cases to ensure that it generated the expected solution. For instance, when the due date of all orders is set to zero and the total items to be collected are less than the AMR cart capacity, the optimal solution is expected to utilize all the available resources and make only one tour for each AMR. Likewise, all orders are likely to be delivered to the depot without any tardiness if their due date is set to be very high. The performance of the proposed MILP model is observed to be as expected for these extreme cases. Table 3.5 summarizes the results of the MILP model. The computing time required to achieve the best solution is referred to as the CPU time. Similar to Scholz et al. (2017), the MILP solver is terminated after 7200 seconds.

The MILP model is able to achieve the optimal solution for all the instances involving 10 and 15 items. As expected, the total tardiness and computing time increases substantially when $|\mathcal{N}|$ is increased from 10 to 15. A similar trend is observed when the human-robot composition or due date tightness (γ) is increased. With respect to the human-robot composition, the reduction in total tardiness appears to be highest when $p_{\max}$ is increased. For example, when $(p_{\max}, r_{\max})$ is changed from (1,1) to (2,1) for $|\mathcal{N}| = 10$, the total tardiness reduces by 60%, 65% and 60% for γ of 0.6, 0.7 and 0.8, respectively. On the other hand, increasing the available AMRs instead of human workers (i.e., adjusting human-robot composition from (1,1) to

Table 3.5: MILP performance

N	(r_max, p_max)	MTCR	MILP	
			tardiness	CPU time
10	(1,1)	0.6	55	14
10	(1,1)	0.7	70	27
10	(1,1)	0.8	118	42
10	(1,2)	0.6	22	51
10	(1,2)	0.7	25	55
10	(1,2)	0.8	47	59
10	(2,1)	0.6	41	30
10	(2,1)	0.7	51	46
10	(2,1)	0.8	87	51
10	(2,2)	0.6	19	60
10	(2,2)	0.7	34	109
10	(2,2)	0.8	71	131
15	(1,1)	0.6	126	190
15	(1,1)	0.7	140	240
15	(1,1)	0.8	177	554
15	(1,2)	0.6	94	759
15	(1,2)	0.7	100	1034
15	(1,2)	0.8	118	2412
15	(2,1)	0.6	106	691
15	(2,1)	0.7	115	1175
15	(2,1)	0.8	140	2465
15	(2,2)	0.6	77	1521
15	(2,2)	0.7	82	1731
15	(2,2)	0.8	98	2847

(1,2)) for $|\mathcal{N}| = 10$ yields 24%, 27%, and 26% improvement in tardiness for γ of 0.6, 0.7 and 0.8, respectively. This finding could be attributed to the slower travel speeds of pickers compared to AMRs. In other words, increasing the resource with slower speeds has the potential to reduce the AMR wait time at a pick location, which, in turn, reduces the completion time of a batch (and associated tardiness).

Chapter 4

Deterministic Local Search Algorithm for Collaborative Order Picking System

The MILP model can produce an optimal solution in a reasonable time for small instances of CHR-OBASRP. However, given the NP-hard nature of the problem under consideration, the optimization model becomes computationally intractable as the problem size increases. To deal with such instances, we propose a simulated annealing approach with adaptive neighborhood search and optimization-based restarts.

4.1 Overview of Variable Neighborhood Descent Approach

The neighborhood descent approach is an uncomplicated deterministic local search approach (Henn, 2015). The approach starts from an initial solution. For each iteration, the approach will calculate the objective value with the selected neighborhood structure. If the neighborhood structure reduces a better objective value, it will be kept and become the initial solution for the next iteration. Otherwise, another neighborhood structure will be selected and tested. The algorithm can be ended after a certen number of iterations, or while the objective value stops improving. The neighborhood structures considered in this research and their adaptive selection procedure are discussed in Section 4.3.

Algorithm 1 Variable Neighborhood Search

1: **Inputs:** problem parameters and N_l, where $l = \{1, 2, ..., l^{\max}\}$
2: generate initial solution (ν^0) and compute corresponding total tardiness $f(\nu^0)$
3: initialize $\nu^* \leftarrow \nu^0$, $f(\nu^*) \leftarrow f(\nu^0)$, and $l \leftarrow 1$
4: **while** $l \leq l^{\max}$ **do**
5: find best neighborhood solution ν' for operator $N_l(\nu)$ and determine $f(\nu')$
6: **if** $f(\nu') < f(\nu^*)$ **then**
7: $\nu^* \leftarrow \nu'$ and $f(\nu^*) \leftarrow f(\nu')$
8: $l = 1$
9: **else**
10: $l \leftarrow l + 1$
11: **end if**
12: **end while**
13: **return** $f(\nu^*)$ and ν^*

The algorithm requires parameters associated with the CHR-OBASRP instance and the different neighborhood operators as inputs (line 1). The initial solution is generated using the constructive heuristic and the corresponding total tardiness is determined (line 2). The best found solution, the best tardiness value, and the current

neighborhood operator are initialized at the beginning of the search procedure (line 3). The loop (lines 4-12) seeks to sequentially explore the best neighborhood solution pertaining to each operator, and terminates the VND procedure only when the current best solution (ν^*) cannot be improved by any of the $l^{\max}$ operators. For the chosen operator l, the algorithm searches for the best neighborhood solution (i.e., lowest total tardiness value). If an improved solution is identified through local search, then it is accepted as the current best solution and the neighborhood operator is reset to 1 (lines 6-8). Otherwise, the next neighborhood operator ($l + 1$) is explored. Upon termination, the best-found solution and corresponding tardiness value are returned (line 13).

4.2 Rule-based Heuristic for Initial Solution Generation

A rule-based constructive approach is used to generate an initial feasible mission and pick lists for the AMRs and workers, respectively. We adapt the earliest start date heuristic proposed by Henn (2015) for a human-only OPS to the CHR-OBASRP. Algorithm 2 provides the procedure for generating the initial solution to the CHR-OBASRP.

Algorithm 2 Rule-based heuristic for CHR-OBASRP

1: **Input:** $\mathcal{N}$, $\mathcal{J}$, $\mathcal{O}_j$,$\mathcal{R}$, $\mathcal{K}$, $\mathcal{T}$, d_j, $\tau^{\mathcal{R}}_{i',i}$, κ_r, $\tau^{\mathcal{K}}_{i',i}$, η^U_i, η^L_i
2: **Output:** mission lists ($\mathcal{M}_{rt}$, $\forall r \in \mathcal{R}, t \in \mathcal{T}$) and pick lists ($\mathcal{P}_k$, $\forall k \in \mathcal{K}$)
3: Set total items assigned for robot r in trip t to zero (i.e., $\bar{I}_{r,t} = 0 \ \ \forall r \in \mathcal{R}, \forall t \in \mathcal{T}$)
4: Initialize $\mathcal{M}_{rt}$, $\forall r \in \mathcal{R}, t \in \mathcal{T}$, $\mathcal{P}_k$, $\forall k \in \mathcal{K}$ and current trip indicator (t^*_r, $\forall r \in \mathcal{R}$)
5: Create list $\mathcal{L}$ by adding items $i \in \mathcal{O}_j$ corresponding to order $j \in \mathcal{J}$ from smallest to largest d_j
6: **while** $\mathcal{L}^{\mathcal{R}} \neq \emptyset$ **do**
7: Select the first item i_1 from $\mathcal{L}$
8: **for** each operator $k \in \mathcal{K}$ **do**
9: Estimate $B^K_{i_1,k}$ and $F^K_{i_1,k}$
10: **end for**
11: Assign item i_1 to operator $k^* = \underset{k \in \mathcal{K}}{\operatorname{argmin}}\{B^K_{i_1,k}\}$
12: Append item i_1 to the end of $\mathcal{P}_{k^*}$
13: **for** each robot $r \in \mathcal{R}$ **do**
14: Select current trip t^*_r
15: **if** $\bar{I}_{r,t} < \kappa_r$ **then**
16: Compute collection begin time for robot r in trip t, $B^R_{i_1,r,t^*_r}$
17: **else**
18: **if** $t^*_r < |\mathcal{T}|$ **then**
19: Set trip indicator $t^*_r = t^*_r + 1$
20: Compute collection begin time for robot r in trip t^*_r, $B^R_{i_1,r,t^*_r}$
21: **else**
22: Set $B^R_{i_1,r,t^*_r} = M$, where M is a large positive number
23: **end if**
24: **end if**
25: **end for**
26: Assign item i_1 to robot $r^* = \underset{r \in \mathcal{R}}{\operatorname{argmin}}\{B^R_{i_1,r,t^*_r}\}$
27: Append item i_1 to the end of $\mathcal{M}_{r^*,t^*_r}$
28: Remove item i_1 from the list $\mathcal{L}$.
29: **end while**
30: **return** $\mathcal{M}_{r,t}$ and $\mathcal{P}_k$

The algorithm uses the problem parameters discussed in Section 3.2 as inputs, and determines the mission and pick lists (lines $1-2$). First, the key variables are initialized, and the items are sorted in ascending order of their due dates (lines $4-5$). Subsequently, each item from the sorted listed $\mathcal{L}$ is sequentially assigned to an operator and robot (lines $6-29$). We compute the time at which each worker can reach the pick location from the last position to begin item retrieval (lines $8-10$), and assign the operator with the earliest start time to pick the item (lines 11 and 12). Similarly, the collection begin time for each AMR is calculated based on its

travel time from the previous location, and the item is assigned to the AMR with the earliest collection begin time (lines $13 - 27$). Since the AMR's cart capacity is limited, the heuristic ensures feasibility before assigning an item to the AMR (line $15 - 24$). Once an item is assigned to an operator and robot, it is removed from the list $\mathcal{L}$ (line 28). This procedure is repeated until all the items in sorted list $\mathcal{L}$ have been removed.

4.3 Neighborhood Structures

The neighborhood operators facilitate the exploration of the search space by generating new solutions given a feasible solution. In this research, we perturb both the mission and pick lists to generate a new neighborhood solution. First, we modify the current solution pertaining to the AMR by considering two commonly used search operators - (i) move (one item is selected from the current solution and shifted to a new position) and (ii) swap (two items from the current solution are selected and interchanged). In particular, we consider the following eight move and swap operators to explore the solution space associated with the AMR.

- N_1: Move an item from a robot tour to a new position in that same tour.
- N_2: A randomly selected item transported by a robot is moved to the tour of another robot

- N_3: The entire batch of items processed in a tour by an AMR is moved to a new position, but still assigned to the same AMR.
- N_4: An item is randomly selected and moved to another tour of the same robot.
- N_5: Swaps two randomly selected items transported by the same robot, but in different tours.
- N_6: Two items assigned to different AMRs are randomly selected and swapped.
- N_7: Two randomly selected tours made by the same robot are interchanged.
- N_8: Swaps two randomly selected items within a robot tour.

Figures 4.1 and 4.2 provide an illustrative example of the different move and swap neighborhood structures, respectively. It is important to note that the eight operators are designed to ensure that all the subproblems associated with the AMR (order batching, tour assignment and sequencing, AMR routing) are feasible when a neighborhood solution is generated. For instance, neighborhood structure N_1 will not move an item to a tour if such transformation will result in a non-empty tour after an empty tour. In other words, the eight neighborhood structures will identify all possible feasible moves/swaps and will randomly select one among them to generate a feasible neighborhood solution.

Subsequently, the current operator plan is modified, while taking into account the new neighborhood solution for the AMR. Otherwise, the solution to the problem can

become infeasible. For example, an AMR might be scheduled to visit items in the following sequence $1 \rightarrow 7 \rightarrow 5$ in tour 1. However, the picker's plan could be items $1 \rightarrow 5 \rightarrow 7$, thereby making the solution to the CHR-OBASRP infeasible since AMR will continue to wait at pick location 7 for the human worker, while the picker will be waiting at pick location 5 for the AMR. The overall solution to the problem can be made feasible in many ways such as swapping items 5 and 7 for the worker or reassigning item 5 or 7 to another picker. Thus, given the new neighborhood solution for the AMR, the current operator plan is updated by identifying all possible items that can be moved or reassigned, and randomly selecting a feasible operation.

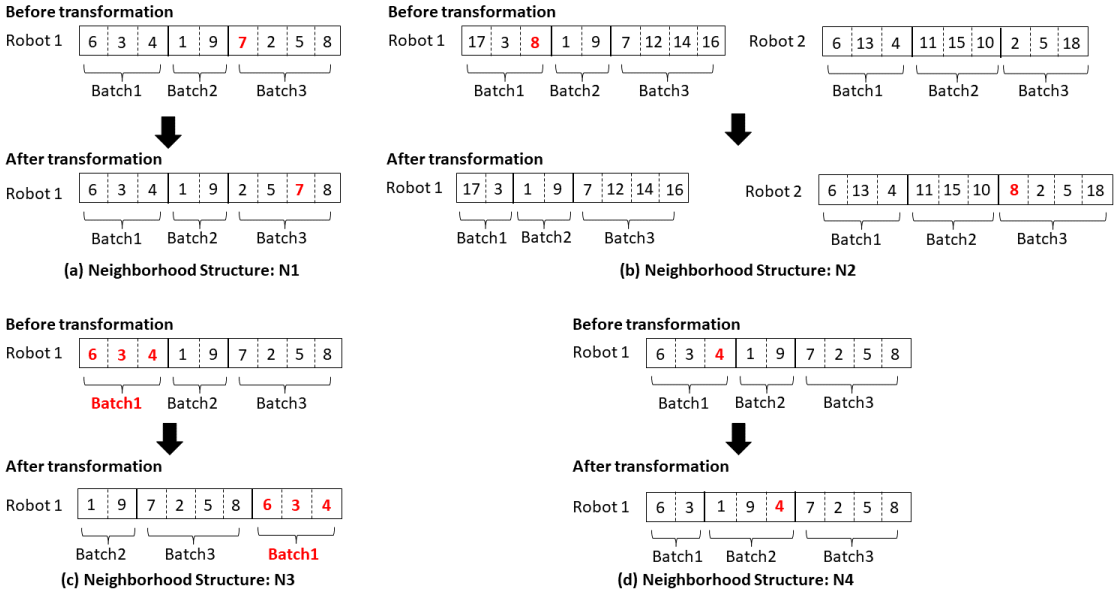


Figure 4.1: Illustration of neighborhood structures that uses move operator ($N_1 - N_4$)

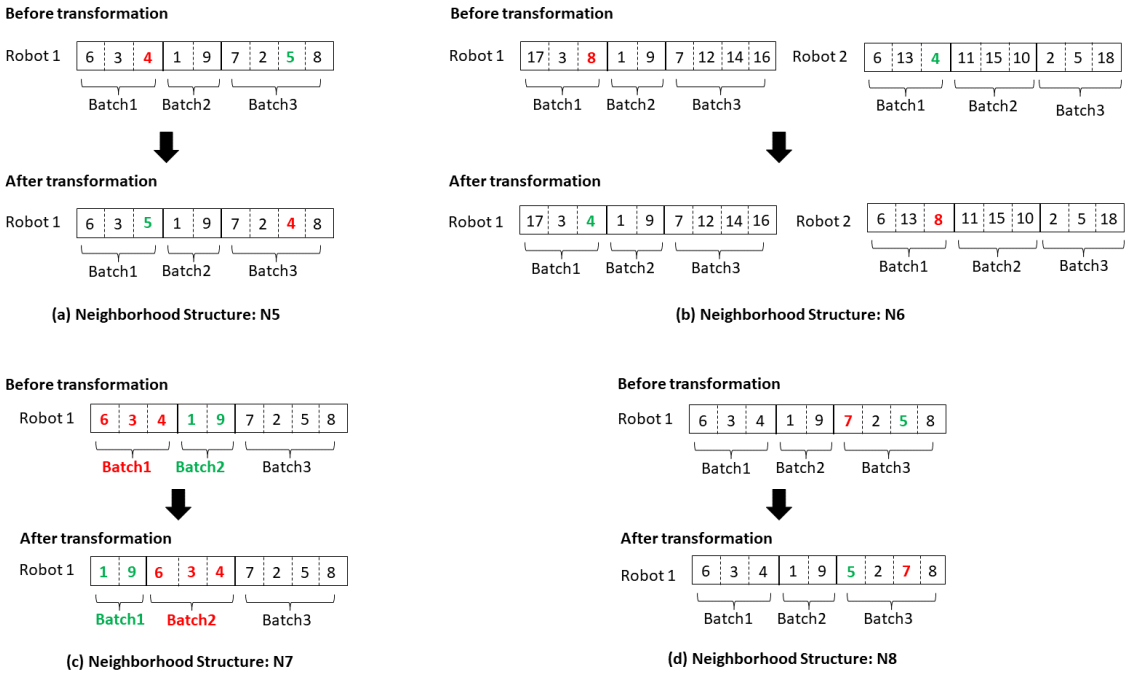


Figure 4.2: Illustration of neighborhood structures that uses swap operator ($N_5 - N_8$)

4.4 Results of Exact and VND Approach

In this section, we solve the proposed MILP model and verify the solution obtained. To assess the accuracy and correctness of the MILP model, we manually compute performance measures based on the optimal value of the decision variables and compare them against the solution obtained. We found the two solutions (manually computed vs. model output) to be in complete agreement for the small instances that are randomly selected. Furthermore, we also evaluated the model for certain edge cases to ensure that it generated the expected solution. For instance, when the due date of all orders is set to zero and the total items to be collected are less than the AMR cart capacity, the optimal solution is expected to utilize all the available

resources and make only one tour for each AMR. Likewise, all orders are likely to be delivered to the depot without any tardiness if their due date is set to be very high. The performance of the proposed MILP model is observed to be as expected for these extreme cases. Subsequently, the VND approach is used for solving all the small instances, and the performance is compared against the optimal solution. Table 5.1 summarizes the results of the MILP model and also provides the relative percentage deviation (RPD) between the solution obtained by VND approach and MILP model. The RPD is calculated as $100 \cdot (Z_s - Z^*)/Z_s$, where Z_s is the best total tardiness achieved by solution approach $s \in \{\mathrm{V}ND\}$ and Z^* is the optimal total tardiness. The computing time required to achieve the best solution is referred to as the CPU time. Similar to Scholz et al. (2017), the MILP solver is terminated after 7200 seconds.

The MILP model is able to achieve the optimal solution for all the instances involving 10 and 15 items. As expected, the total tardiness and computing time increases substantially when $|\mathcal{N}|$ is increased from 10 to 15. A similar trend is observed when the human-robot composition or due date tightness (γ) is increased. With respect to the human-robot composition, the reduction in total tardiness appears to be highest when $p_{\max}$ is increased. For example, when $(p_{\max}, r_{\max})$ is changed from (1,1) to (2,1) for $|\mathcal{N}| = 10$, the total tardiness reduces by 60%, 65% and 60% for γ of 0.6, 0.7 and 0.8, respectively. On the other hand, increasing the available AMRs instead of human workers (i.e., adjusting human-robot composition from (1,1) to

Table 4.1: Performance comparison of metaheuristic approach against the exact approach

N	(r_max, p_max)	MTCR	MILP		VND	
			tardiness	CPU time	RPD	CPU time
10	(1,1)	0.6	55	14	1.0%	18
10	(1,1)	0.7	70	27	1.2%	20
10	(1,1)	0.8	118	42	0%	27
10	(1,2)	0.6	22	51	2.0%	36
10	(1,2)	0.7	25	55	0.5%	41
10	(1,2)	0.8	47	59	0.8%	53
10	(2,1)	0.6	41	30	0%	28
10	(2,1)	0.7	51	46	0.7%	37
10	(2,1)	0.8	87	51	1.3%	41
10	(2,2)	0.6	19	60	0.9%	48
10	(2,2)	0.7	34	109	1.3%	64
10	(2,2)	0.8	71	131	0.8%	92
15	(1,1)	0.6	126	190	1.0%	74
15	(1,1)	0.7	140	240	1.7%	83
15	(1,1)	0.8	177	554	0.5%	88
15	(1,2)	0.6	94	759	0.2%	136
15	(1,2)	0.7	100	1034	0.3%	151
15	(1,2)	0.8	118	2412	0%	163
15	(2,1)	0.6	106	691	1.3%	138
15	(2,1)	0.7	115	1175	1.0%	155
15	(2,1)	0.8	140	2465	0.8%	164
15	(2,2)	0.6	77	1521	0.9%	168
15	(2,2)	0.7	82	1731	1.3%	179
15	(2,2)	0.8	98	2847	1.6%	183

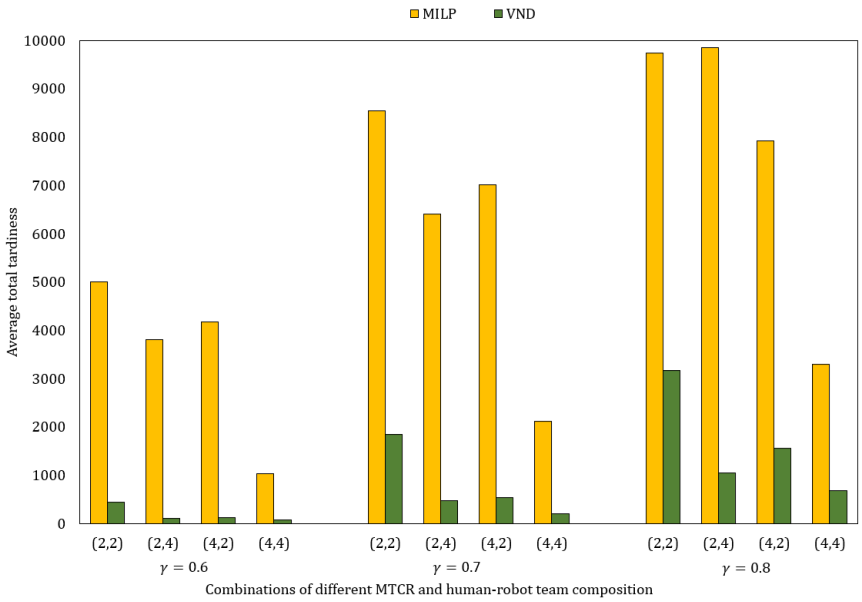


Figure 4.3: Impact of varying AMR cart capacity on average total tardiness for instances with $|\mathcal{N}| = 50$

(1,2)) for $|\mathcal{N}| = 10$ yields 24%, 27%, and 26% improvement in tardiness for γ of 0.6, 0.7 and 0.8, respectively. This finding could be attributed to the slower travel speeds of pickers compared to AMRs. In other words, increasing the resource with slower speeds has the potential to reduce the AMR wait time at a pick location, which, in turn, reduces the completion time of a batch (and associated tardiness).

Our empirical analysis also revealed that the computing time of the MILP model increases exponentially as $|\mathcal{N}|$ increases and the MILP solver is unable to find the optimal solution for $|\mathcal{N}| = 20$ within the runtime threshold of 7200 seconds. Figure 5.1 compares the solution obtained by the MILP model (after 7200 seconds) with the VND approach. The time taken by VND to obtain the least tardiness values is less than 1800 seconds. It can be observed that the MILP model produced solutions of

inferior quality. Thus, similar to prior literature on OBASRP (Scholz et al. (2017)), we conclude that the exact approach is not suitable for obtaining good solutions in practive, especially when $|\mathcal{N}|$ exceeds 20.

Table 4.2: Performance of VND approach

n	**(r_max, p_max)**	**MTCR**	**VND**
50		0.6	438
50	(2,2)	0.7	1848
50		0.8	3168
50		0.6	112
50	(2,4)	0.7	471
50		0.8	1045
50		0.6	131
50	(4,2)	0.7	538
50		0.8	1564
50		0.6	77
50	(4,4)	0.7	208
50		0.8	674
100		0.6	792
100	(2,2)	0.7	3999
100		0.8	10438
100		0.6	313
100	(2,4)	0.7	1137
100		0.8	3431
100		0.6	354
100	(4,2)	0.7	1494
100		0.8	5422
100		0.6	214
100	(4,4)	0.7	507
100		0.8	2362

The performance of VND for the problem class pertaining to large instances ($|\mathcal{N}| =$ 50 and 100) is presented in Table 4.2. Similar to the findings for small instances, the total tardiness for all problem class increases substantially with γ. Also, increasing the availability of relatively slower-speed resources (i.e., pickers) yields the highest reduction in average total tardiness.

Chapter 5

Stochastic Metaheuristic Approach for Collaborative Order Picking System

A deterministic heuristic approach (i.e., VND) was developed and evaluated in Chapter 4. However, a potential limitation of VND is that it can get stuck in a local optimal solution and may not yield good quality solutions for certain instances. Therefore, in this chapter, new variants of simulated annealing, a stochastic metaheuristic approach, will be developed and evaluated to effectively solve large instances of the CHR-OBASRP.

5.1 Overview of Proposed Simulated Annealing Approach

Simulated annealing (SA) is a metaheuristic technique that is often employed to efficiently solve NP-hard combinatorial optimization problems. It is based on the physical annealing process, where the microstructure of a material is altered by gradually cooling it from a higher temperature. In particular, SA uses a local neighborhood search procedure to explore the solution space to the problem, where the search process is governed by the following key parameters - initial temperature ($\theta^{\max}$), cooling rate (α), and final temperature ($\theta^{\min}$).

The algorithm begins with an initial solution ν^0 at temperature $\theta^{\max}$, and the associated objective function value is $f(\nu^0)$. In the case of CHR-OBASRP, a solution refers to the mission and pick list for all the AMRs and workers, respectively ($\mathcal{M}_{rt},\ r \in \mathcal{R}, t \in \mathcal{T}$ and $\mathcal{P}_k,\ k \in \mathcal{K}$), while the objective function value corresponds to the total tardiness $\sum_{j \in \mathcal{J}} T_j$. The representation of the solution to the CHR-OBASRP is discussed in previous section. To initiate the iterative search process, the current solution (ν), objective ($f(\nu)$) and temperature (θ) values are set as ν^0, $f(\nu^0)$, $\theta^{\max}$, respectively. In addition, the best found solution is also initialized ($\nu^* \leftarrow \nu^0$, $f(\nu^*) \leftarrow f(\nu^0)$). At a given iteration I and temperature θ, a candidate solution ν' with objective $f(\nu')$ is generated by exploring the neighborhood solutions of ν. For a minimization objective, if $f(\nu') \leq f(\nu)$, then the neighboring solution is set as the incumbent and best found solution (i.e., $\nu \leftarrow \nu'$ and $\nu^* \leftarrow \nu'$). Otherwise, the

inferior neighborhood solution is accepted as the incumbent with a certain likelihood. A common acceptance probability function for the SA algorithm corresponds to the metropolis criterion, where a worse solution is accepted as incumbent if a randomly generated number $g \sim (0,1) < e^{(f(\nu)-f(\nu'))/\theta}$. The procedure of exploring the neighborhood solutions and updating the incumbent accordingly is repeated ρ times at each θ. Subsequently, the temperature is reduced by a cooling rate of α (i.e., $\theta \leftarrow \alpha \times \theta$). The algorithm is terminated and the best found solution along with the objective is returned when one of the following criteria is satisfied (i) temperature reaches below $\theta^{\min}$, (ii) $f(\nu)$ remains unchanged for the $\rho^{\max}$ iterations of the search procedure (iii) $f(\nu)$ is 0 (since total tardiness cannot be lower than 0). SA allows the exploration of diverse regions by accepting inferior solutions with a probability that decreases steadily with the temperature. It also achieves intensification by always accepting improved solutions.

In this research, we propose a variant of the traditional SA algorithm called the restarted simulated annealing with adaptive neighborhood search (RSA-ANS). While SA achieves a trade-off between feasible region exploration and exploitation, the large search space arising due to the three subproblems in CHR-OBASRP requires better strategies for escaping local minima and selecting neighborhood operators. Therefore, unlike the traditional SA, the proposed RSA-ANS uses a restart strategy to escape local optima and adaptive neighborhood selection mechanism for achieving time-performance trade-off. The procedure for RSA-ANS is given in Algorithm 3.

The RSA-ANS requires a set of neighborhood operators along with other parameters in traditional SA as inputs (line 1). A feasible initial solution is generated using a constructive heuristic (see Section **??**), and is set as the current as well as best found solutions (lines 2-3). Also, the current temperature and the counter that tracks the total continuous iterations with no improvement (ρ) are initialized (line 3). Subsequently, the total tardiness associated with the current and best-found solutions is determined (line 4). The outer loop of the RSA-ANS (lines 5-39) forms the core of the local search procedure that progresses by gradually lowering the temperature until one of the termination conditions is met. The inner loop (8-30) is repeated $I^{\max}$ times at a given temperature θ, and performs an adaptive neighborhood search to update the current solution.

The neighborhood structures considered in this research and their adaptive selection procedure are discussed in Sections **??** and 5.2, respectively. In particular, at every iteration, the algorithm selects π operators from $l^{\max}$ structures and generates neighborhood solutions using them (lines 9-10). The neighborhood solution that yields the lowest tardiness is then selected (lines 11). The current solution is replaced with the best neighborhood solution as per the traditional SA algorithm (lines 13-28). The notable difference is that we use the relative difference between the current and neighborhood solution in the metropolis acceptance criterion as opposed to the absolute difference (line 16), thereby making it dimensionless and independent of the problem specifications (Parthasarathy & Rajendran, 1997). If the solution does not

change for ρ^{RS} iterations, then the RSA-ANS employs the optimization-based restart strategy (lines 32-38), which is described in Section 5.3. Finally, the RSA-ANS returns the best-found solution and the corresponding tardiness value (line 40).

Algorithm 3 Restarted Simulated Annealing with Adaptive Neighborhood Search

Inputs: problem parameters, θ^0, $\theta^{\min}$, α, $I^{\max}$, $\rho^{\max}$, ρ^{RS}, π, N_l where $l = \{1, 2, ..., l^{\max}\}$
generate initial solution (i.e., mission and pick lists), ν^0
initialize $\nu \leftarrow \nu^0$, $\nu^* \leftarrow \nu^0$, $\theta \leftarrow \theta^0$, $\rho \leftarrow 0$
compute $f(\nu)$ and $f(\nu^*)$
while $\theta > \theta^{\min}$ **and** $\rho < \rho^{\max}$ **and** $f(\nu^*) \neq 0$ **do**
    $I \leftarrow 1$
    $p_l = \dfrac{1}{l^{\max}}, \forall l = \{1, 2, ..., l^{\max}\}$
    while $I \leq I^{\max}$ **and** $f(\nu^*) \neq 0$ **do**
        choose π operators from $\{N_1, N_2, ..., N_{l^{\max}}\}$ based on selection probability p_l
        generate neighborhood solutions, $\nu_n, n = 1, 2, ..., \pi$, using selected operators
        compute tardiness of all neighborhood solutions, $f(\nu_n), n = 1, 2, ..., \pi$
        $f(\nu') \leftarrow \min\{f(\nu_1), f(\nu_2), ..., f(\nu_\pi)\}$ and $\nu' \leftarrow \nu'_{\text{argmin}\,\{f(\nu_1), f(\nu_2), ..., f(\nu_\pi)\}}$
        if $f(\nu') < f(\nu)$ **then**
            $\nu \leftarrow \nu'$ and $f(\nu) \leftarrow f(\nu')$
        else
            $\Delta = \dfrac{f(\nu') - f(\nu)}{f(\nu)}$
            generate random number $g \sim U(0, 1)$
            if $g \leq e^{-\Delta/\theta}$ **then**
                update $\nu \leftarrow \nu'$ and $f(\nu) \leftarrow f(\nu')$
            end if
            if $f(\nu') < f(\nu^*)$ **then**
                $\nu^* \leftarrow \nu'$ and $f(\nu^*) \leftarrow f(\nu')$
                $\rho \leftarrow 0$
            else
                $\rho \leftarrow \rho + 1$
            end if
            adjust p_l of neighborhood operator N_l, where $l = \{1, 2, ..., l^{\max}\}$ (see Section **??**)
        end if
        $I \leftarrow I + 1$
    end while
    $\theta \leftarrow \alpha \times \theta$
    if $\rho \geq \rho^{\text{RS}}$ **then**
        Use *fix-and-optimize* heuristic to find a new restart point ν'' (see Section 5.3)
        if $f(\nu'') \neq f(\nu)$ **then**
            $\nu \leftarrow \nu''$ and $f(\nu) \leftarrow f(\nu'')$
            $\rho \leftarrow 0$
        end if
    end if
end while
return $f(\nu^*)$ and ν^*

5.2 Adaptive Selection of Neighborhood Operators

Given a set of $l^{\max}$ neighborhood structures $(N_1, N_2, ..., N_{l^{\max}})$, the RSA-ANS selects π operators (where $\pi \leq l^{\max}$) to generate multiple neighborhood solutions. Evaluating all the operators at each iteration (i.e., $\pi = l^{\max}$) may allow the exploration of many neighboring solutions, but increases the computationally complexity as $l^{\max}$ increases. On the other hand, randomly selecting a few neighborhood operators at every iteration may not efficiently guide the search toward promising solution regions. Therefore, to achieve a performance-efficiency tradeoff, it is important to strategically select the operators at a given iteration. The success of a neighborhood structure depends on several factors such as problem instances and current solutions. Therefore, a neighborhood operator could achieve improved objective value during initial iterations of the local search procedure, but may not find good solutions as the algorithm progresses. Therefore, the only mechanism to select a well-performing neighborhood operator at a given temperature is through dynamic adjustments of their selection probability.

In this research, we adaptively select π operators based on their successful performance in previous iterations. A selection weight w_l is assigned for each operator and is initialized to be the same for all operators. Subsequently, the weights are updated at every iteration of a given temperature according to Equation (5.1), where ξ denotes the minimum weight assigned to all operators and ϕ_l represents the number of

iterations in which the selected operator was accepted as the current solution.

$$w_l = \xi + (1 - l^{\max} \times \xi) \times \frac{\phi_l}{\sum_l \phi_l} \qquad \forall l = 1, 2, ..., l^{\max} \tag{5.1}$$

To select the π operators in an iteration, a roulette-wheel selection principle is used, where a neighborhood operator N_l has a selection probability p_l, as given by Equation (5.2). Note that the operators are selected without replacement, and therefore, the same operator cannot be chosen twice.

$$p_l = \frac{w_l}{\sum_l w_l} \qquad \forall l = 1, 2, ..., l^{\max} \tag{5.2}$$

In order to randomize the search procedure and provide an equal chance to all operators in the later stages, the weights and selection probabilities are reset after γ temperature reductions.

5.3 Fix-and-Optimize Heuristic as Restart Strategy

During the search process, if $f(\nu)$ does not change for $\rho^{\max}$ continuous temperature reductions, then we adopt a fix-and-optimize heuristic strategy to escape local minima and update the current solution ν. The fix-and-optimize heuristic seeks to fix specific binary variables in the MILP model, and solve for the remaining decision variables to obtain a new solution. We consider the following two cases for fixing the binary variables.

- **Case 1:** The binary variables associated with the pick list $\{\mathcal{P}_k, k \in \mathcal{K}\}$ of the current solution ν are fixed and the mission list $\{\mathcal{M}_{rt}, r \in \mathcal{R}, t \in \mathcal{T}\}$ of ν is used to warm-start the model. Subsequently, the MILP model is solved to optimize the mission list $\{\mathcal{M}_{rt}, r \in \mathcal{R}, t \in \mathcal{T}\}$. This will lead to a new solution ν' with an updated mission list $\{\mathcal{M}_{rt}^{new}, r \in \mathcal{R}, t \in \mathcal{T}\}$ and objective value $f(\nu')$.
- **Case 2:** The binary decision variables pertaining to $\{\mathcal{M}_{rt}, r \in \mathcal{R}, t \in \mathcal{T}\}$ of ν are fixed, while $\{\mathcal{P}_k, k \in \mathcal{K}\}$ is used to warm-start and solve the MILP model. This results in updated solution ν' with objective value $f(\nu')$ and new pick list $\{\mathcal{P}_k^{new}, k \in \mathcal{K}\}$.

To restart the algorithm, one of the two cases is randomly selected to replace the current solution ν and the corresponding objective value with ν' and $f(\nu')$, respectively. The potential advantages of the proposed restart mechanism are as follows.

First, a new solution can be found even for very large instances by terminating the optimization model after a certain time limit. In other words, it is not necessary to solve the MILP model to optimality, especially when it is expected to substantially slow the computational time for RSA-ANS. Second, as opposed to a random restart, the proposed approach is guaranteed to yield an objective value that is better than or equal to the current $f(\nu)$. Finally, leveraging optimization model is likely to direct the restart point to more promising solution regions.

5.4 Performance Evaluation of RSA-ANS

The solution quality obtained by the proposed RSA-ANS is evaluated by comparing it with the variable neighborhood descent (VND) algorithm - a metaheuristic approach that was known to provide high-quality solutions to the OBASRP problem for a conventional human-only OPS in the literature (Henn, 2015; Scholz et al., 2017). An overview of the VND procedure is given in Algorithm 4. The algorithm requires parameters associated with the CHR-OBASRP instance and the different neighborhood operators as inputs (line 1). Similar to RSA-ANS, the initial solution is generated using the constructive heuristic and the corresponding total tardiness is determined (line 2). The best found solution, the best tardiness value, and the current neighborhood operator are initialized at the beginning of the search procedure (line 3). The loop (lines 4-12) seeks to sequentially explore the best neighborhood

solution pertaining to each operator, and terminates the VND procedure only when the current best solution (ν^*) cannot be improved by any of the $l^{\max}$ operators. For the chosen operator l, the algorithm searches for the best neighborhood solution (i.e., lowest total tardiness value). If an improved solution is identified through local search, then it is accepted as the current best solution and the neighborhood operator is reset to 1 (lines 6-8). Otherwise, the next neighborhood operator ($l+1$) is explored. Upon termination, the best-found solution and corresponding tardiness value are returned (line 13). In addition to VND, we also consider the simulated annealing with adaptive neighborhood search (SA-ANS), which follows the same procedure as the proposed RSA-ANS but without the restart strategy. This would allow us to investigate the impact of using an optimization-based restart strategy on the solution quality.

Algorithm 4 Variable Neighborhood Search

1: **Inputs:** problem parameters and N_l, where $l = \{1, 2, ..., l^{\max}\}$
2: generate initial solution (ν^0) and compute corresponding total tardiness $f(\nu^0)$
3: initialize $\nu^* \leftarrow \nu^0$, $f(\nu^*) \leftarrow f(\nu^0)$, and $l \leftarrow 1$
4: **while** $l \leq l^{\max}$ **do**
5: find best neighborhood solution ν' for operator $N_l(\nu)$ and determine $f(\nu')$
6: **if** $f(\nu') < f(\nu^*)$ **then**
7: $\nu^* \leftarrow \nu'$ and $f(\nu^*) \leftarrow f(\nu')$
8: $l = 1$
9: **else**
10: $l \leftarrow l + 1$
11: **end if**
12: **end while**
13: **return** $f(\nu^*)$ and ν^*

5.5 Results of Exact and Metaheuristic Approach Applied to Small Instances

In this section, we compared the performance of SA-ANS and RSA-ANS with MILP and VND model mentioned last chapter. Table 5.1 summarizes the results of the MILP model and also provides the relative percentage deviation (RPD) between the solution obtained by each metaheuristic approach and MILP model. Same as last chapter, the RPD is calculated as $100 \cdot (Z_s - Z^*)/Z_s$, where Z_s is the best total tardiness achieved by solution approach $s \in \{\text{VND, SA-ANS, RSA-ANS}\}$ and Z^* is the optimal total tardiness. The computing time required to achieve the best solution is referred to as the CPU time. Similar to Scholz et al. (2017), the MILP solver is terminated after 7200 seconds.

The results in Table 5.1 indicate superior performance of the three metaheuristic approaches as the average RPD from optimal is less than 1%, and the CPU time is considerably less than the exact approach. Among the three metaheuristics, the proposed RSA-ANS achieved the best tardiness value since it yielded the optimal solution in most cases (14 out of 24 problem classes) and resulted in the lowest average RPD of 0.05%. The SA-ANS and VND have an average RPD of 0.33% and 0.875%, respectively. This suggests that the proposed fix-and-optimize restart strategy in RSA-ANS has enabled it to achieve better solution quality. Since RSA-ANS has

Table 5.1: Performance comparison of metaheuristic approaches against the exact approach

$\|\mathcal{N}\|$	$(p_{\max}, r_{\max})$	γ	**MILP**		**RPD (CPU time)**		
			Z^*	**CPU time**	**VND**	**SA-ANS**	**RSA-ANS**
10	(1,1)	0.6	55	15	1.0% (13)	0.4% (14)	0% (14)
10	(1,1)	0.7	70	27	1.2% (20)	0.9% (21)	0.1% (25)
10	(1,1)	0.8	118	42	0% (27)	0% (26)	0% (31)
10	(2,1)	0.6	22	51	2% (36)	0.2% (33)	0% (40)
10	(2,1)	0.7	25	55	0.5% (41)	0% (46)	0% (51)
10	(2,1)	0.8	47	59	0.8% (53)	0.3% (50)	0.1% (54)
10	(1,2)	0.6	41	30	0% (28)	0% (30)	0% (30)
10	(1,2)	0.7	51	46	0.7% (37)	0% (39)	0% (39)
10	(1,2)	0.8	87	51	1.3% (41)	0.3% (47)	0.2% (48)
10	(2,2)	0.6	19	60	0.9% (48)	0.7% (49)	0.1% (53)
10	(2,2)	0.7	34	109	1.3% (64)	0.9% (69)	0.1% (70)
10	(2,2)	0.8	71	131	0.8% (92)	0.3% (99)	0% (101)
15	(1,1)	0.6	126	190	1% (74)	0.3% (76)	0% (79)
15	(1,1)	0.7	140	240	1.7% (83)	0.2% (84)	0.2% (86)
15	(1,1)	0.8	177	554	0.5% (88)	0% (90)	0.1% (91)
15	(2,1)	0.6	94	759	0.2% (136)	0% (141)	0% (147)
15	(2,1)	0.7	100	1034	0.3% (151)	0.3% (167)	0% (173)
15	(2,1)	0.8	118	2412	0% (163)	0.7% (169)	0% (179)
15	(1,2)	0.6	106	691	1.3% (138)	0% (145)	0.05% (146)
15	(1,2)	0.7	115	1175	1.0% (155)	0% (160)	0.2% (173)
15	(1,2)	0.8	140	2465	0.8% (164)	1.3% (189)	0% (192)
15	(2,2)	0.6	77	1521	0.9% (168)	0.9% (193)	0% (196)
15	(2,2)	0.7	82	1731	1.3% (179)	0% (200)	0% (206)
15	(2,2)	0.8	98	2847	1.6% (183)	0.8% (201)	0.05% (210)

an additional module pertaining to the restart strategy, its CPU time is slightly higher than SA-ANS and VND. Thus, the analysis of small instances demonstrates the ability to achieve near-optimal solutions by all three algorithms, and highlights the dominance of RSA-ANS. Our empirical analysis also revealed that the computing time of the MILP model increases exponentially as $|\mathcal{N}|$ increases and the MILP solver is unable to find the optimal solution for $|\mathcal{N}| = 20$ within the runtime threshold of 7200 seconds.

Figure 5.1 compares the solution obtained by the MILP model (after 7200 seconds) with the three metaheuristic approaches. The time taken by VND, SA-ANS, and

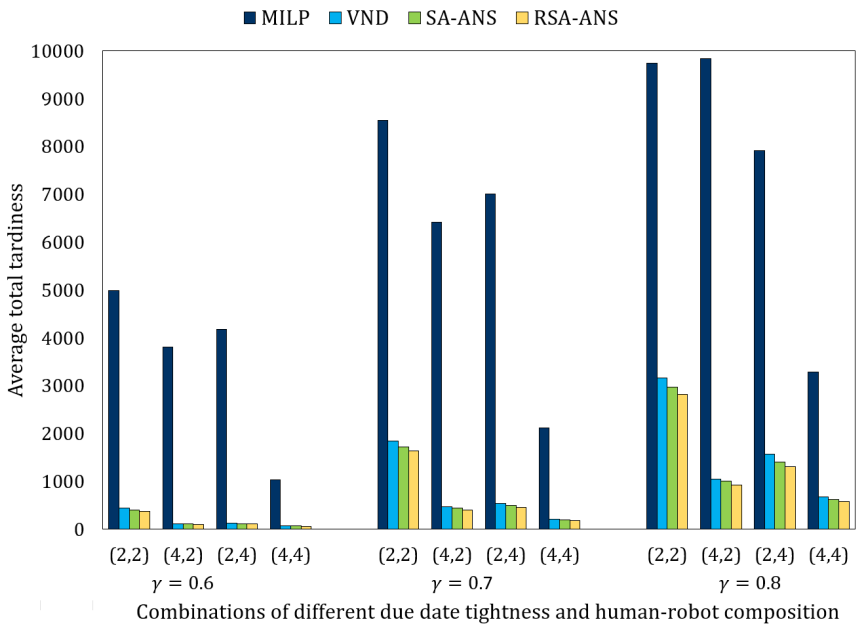


Figure 5.1: Impact of varying AMR cart capacity on average total tardiness for instances with $|\mathcal{N}| = 50$

RSA-ANS to obtain the least tardiness values is less than 1800 seconds. It can be seen that the MILP model produced solutions of inferior quality. For example, the RSA-ANS algorithm achieves improvements ranging from 71% ($\gamma = 0.8$ and $(p_{\max}, r_{\max}) = (2, 2)$) to 91% ($\gamma = 0.6$ and $(p_{\max}, r_{\max}) = (4, 2)$). Thus, similar to prior literature on OBASRP (Scholz et al., 2017), we conclude that the exact approach is not suitable for obtaining good solutions in practice, especially when $|\mathcal{N}|$ exceeds 20.

5.6 Performance Evaluation of Proposed RSA-ANS for Large Instances

The performance of VND, SA-ANS and RSA-ANS for the problem class pertaining to large instances ($|\mathcal{N}| = 50$ and 100) is presented in Table 5.2. In includes the average total tardiness for the CHR-OBASRP obtained by the three metaheuristic approaches, and the solution quality of the proposed RSA-ANS. Specifically, Δ_{VND} provides the percentage improvement achieved by RSA-ANS over VND, while $\Delta_{\text{SA-ANS}}$ demonstrates the benefit of employing the fix-and-optimize restart strategy. Similar to the findings for small instances, the total tardiness for all problem class increases substantially with γ. Also, increasing the availability of relatively slower-speed resources (i.e., pickers) yields the highest reduction in average total tardiness.

It is also evident that RSA-ANS obtains the lowest average total tardiness for all the problem settings under consideration. Compared to VND, the proposed RSA-ANS achieves an average improvement (Δ_{VND}) of 13.25% and 12.70% for instances with 50 and 100 items, respectively. With respect to the four robot-human compositions, the average reduction in total tardiness is consistent and ranges between 12% and 14%. For γ of 0.6, 0.7 and 0.8, the RSA-ANS outperforms VND by 12.57%, 12.77% and 13.57%, respectively. Thus, it is apparent that the proposed RSA-ANS is able to outperform the VND algorithm, which was shown to yield good solution

quality in the literature for a conventional human-only OBASRP (Henn, 2015; Scholz et al., 2017), for all large instances.

The results also establish the significance of the fix-and-optimize restart strategy. As substantiated by $\Delta_{\text{SA-ANS}}$, the average total tardiness obtained without the proposed restart mechanism (ie., SA-ANS) is 6.86% and 7.70% higher for problem instances with $|\mathcal{N} = 50|$ and 100, respectively. For different values of both $(p_{\max}, r_{\max})$ and γ, the total tardiness obtained by RSA-ANS is consistently between 7-8% lower than SA-ANS.

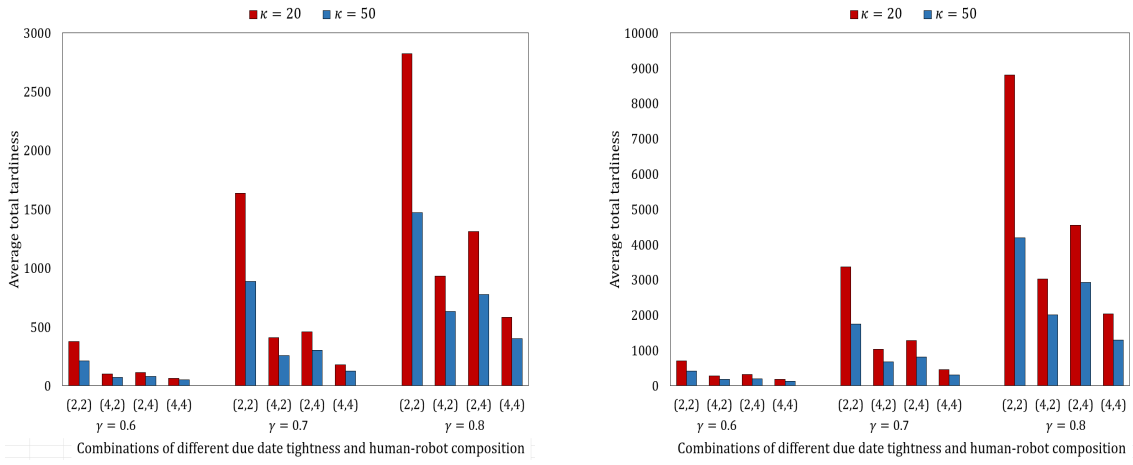
Table 5.2: Average total tardiness for large instances, and solution quality of proposed RSA-ANS

$\lvert\mathcal{N}\rvert$	$(p_{\max}, r_{\max})$	γ	**VND**	**SA-ANS**	**RSA-ANS**	Δ_{VND}	$\Delta_{\text{SA-ANS}}$
50	(2,2)	0.6	438	408	376	14.17%	7.85%
50		0.7	1848	1720	1637	11.42%	4.85%
50		0.8	3168	2973	2825	10.82%	4.98%
50	(4,2)	0.6	112	108	101	10.43%	6.92%
50		0.7	471	438	409	13.14%	6.65%
50		0.8	1045	1009	931	10.83%	7.72%
50	(2,4)	0.6	131	121	113	14.39%	7.01%
50		0.7	538	494	459	14.56%	7.09%
50		0.8	1564	1407	1313	16.06%	6.68%
50	(4,4)	0.6	77	69	65	15.52%	5.97%
50		0.7	208	198	179	13.99%	9.58%
50		0.8	674	626	582	13.66%	6.99%
100	(2,2)	0.6	792	772	706	10.90%	8.66%
100		0.7	3999	3653	3366	15.83%	7.86%
100		0.8	10438	9594	8810	15.60%	8.16%
100	(4,2)	0.6	313	299	283	9.58%	5.35%
100		0.7	1137	1115	1032	9.23%	7.42%
100		0.8	3431	3294	3027	11.77%	8.10%
100	(2,4)	0.6	354	343	318	10.17%	7.29%
100		0.7	1494	1414	1286	13.92%	9.08%
100		0.8	5422	4821	4555	15.98%	5.52%
100	(4,4)	0.6	214	196	181	15.42%	7.65%
100		0.7	507	493	456	10.04%	7.63%
100		0.8	2362	2248	2033	13.93%	9.59%

5.7 Impact of AMR Cart Capacity

The capacity of the AMR cart (κ) was restricted to 20 items in the baseline analysis of large instances. To assess its sensitivity to total tardiness value, we increase κ to 50 and obtain the solution for the large problem class using the proposed RSA-ANS. Figures 5.2(a) and 5.2(b) illustrates the impact of AMR cart capacity for instances with 50 and 100 items, respectively. It is clear that the average total tardiness decreases considerably as the AMR cart capacity is increased. The percentage reduction in total tardiness increases as due date tightness is increased (i.e., varying γ from 0.6 to 0.8). Specifically, in the case of $|\mathcal{N}| = 50$, increasing the AMR cart capacity from 20 to 50, improves the average total tardiness by 49%, 60%, and 64% for γ of 0.6, 0.7 and 0.8, respectively. The improvement achieved is even greater for instances involving 100 items - 54% 62% 68% for γ of 0.6, 0.7 and 0.8, respectively. Likewise, increasing the AMR cart capacity also improved the total tardiness for the different robot-human team compositions to varying degrees. The reduction in total tardiness is highest when $(p_{\max}, r_{\max})$ is (2,2). It is about 85% and 90% for instances involving 50 and 100 items, respectively. Nevertheless, if $(p_{\max}, r_{\max})$ is set as (4,2) and (2,4), the average improvements is around 51% and 56% (for both 50 and 100 item instances), respectively.

On the other hand, when the available robots and pickers are increased to 4, the improvement in total tardiness is reduced to 38% and 45% for $|\mathcal{N}|$ of 50 and 100,



(a) (b)

Figure 5.2: Impact of varying AMR cart capacity on average total tardiness for instances with (a) 50 and (b) 100 items

respectively. Overall, it is observed that increasing the AMR cart capacity from 20 to 50 is beneficial, especially for instances with 100 items, tighter due dates, or fewer AMRs. This could be attributed to the number of tours made by the AMRs. With a higher cart capacity, each AMR would have to make fewer trips back to the depot before completing all the orders, which, in turn, decreases the total tardiness. Furthermore, we also investigated the number of 20-bin AMRs and pickers required to achieve the same level of performance yielded by employing 50-bin AMRs. As expected, more resources are needed when employing low-capacity AMRs. For both $|\mathcal{N}| = 50$ and 100, three 20-bin AMRs and four pickers are required to achieve similar results obtained by two 50-bin AMRs and two pickers. Likewise, the performance achieved by four 20-bin AMRs and four pickers is similar to that of two 50-bin AMRs and three pickers. Thus, given the AMR capacity, the proposed solution approach can also be used to determine the human-robot composition required to achieve a

specific service level.

5.8 Impact of AMR Speed

The initial analysis considered the AMR travel speed to be 2 ft / sec according to related prior literature (Lee & Murray, 2019). In this section, we analyze the impact of AMR travel speed on overall order picking performance. We evaluate the percentage change in the average total tardiness when the AMR speed is changed (decreased and increased) by 1 ft/sec from its baseline value.

Figure 5.3(a) illustrates the percentage increase in the average total tardiness for all large problem classes when the AMR speed is decreased from 2 ft/sec to 1 ft/sec (i.e., 50% reduction). In other words, in this setting, both picker and AMR travel at the same speed. It can be observed that the average tardiness increases substantially (min: 61%, average: 108%, max: 176%) for all the problem classes under consideration. Its impact is slightly higher for instances with 100 items, and this is expected as AMRs may have to make more tours to process all orders. Additionally, the percentage increase in the average total tardiness gradually decreases as the due date tightness (γ) is increased. With regards to human-robot team composition, the highest percentage increase in the tardiness occurs when $(p_{\max}, r_{\max})$ is (4,2), where the values obtained are about $2.5-3.5$ times more than the baseline setting. On the other hand, settings

with 4 robots and 2 pickers had the lowest impact on tardiness. For cases with an equal number of pickers and robots, the percentage increase was about two times the baseline values.

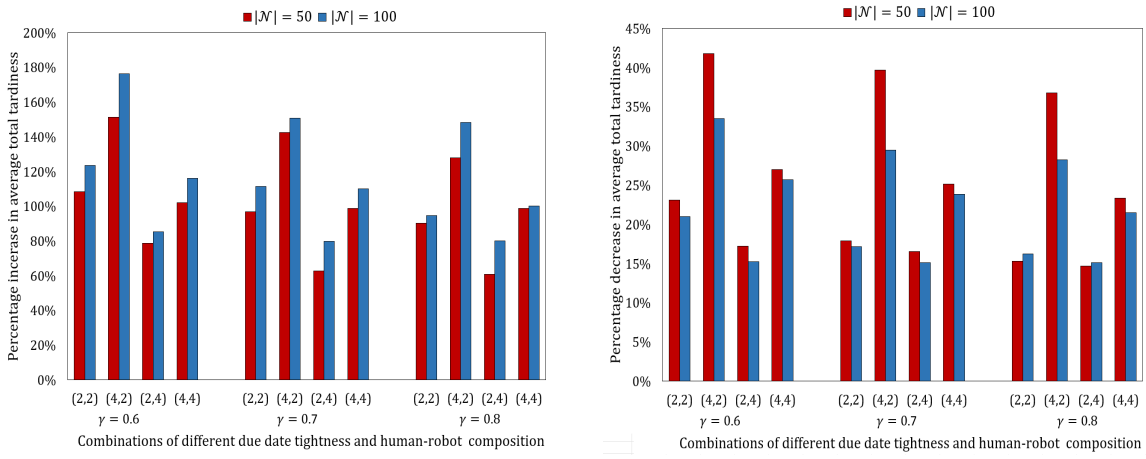


Figure 5.3: Impact of (a) decreasing AMR travel speed from 2 ft/sec to 1 ft/sec and (b) increasing AMR travel speed from 2 ft/sec to 3 ft/sec

Figure 5.3(b) shows the impact of increasing AMR speed from 2 ft/sec to 3 ft/sec. Note that the AMR now travels three times faster than the picker. Although the average total tardiness improves compared to the baseline results (i.e., Table 5.2), the percentage change is not substantial (min: 15%, average: 23%, max: 42%) given that the speed is increased by 50%. The improvement achieved is slightly lower for settings with 100 items or higher due-date tightness. A collaboration among four pickers and two AMRs achieves the highest improvement, whereas employing fewer AMRs than pickers led to the least improvement.

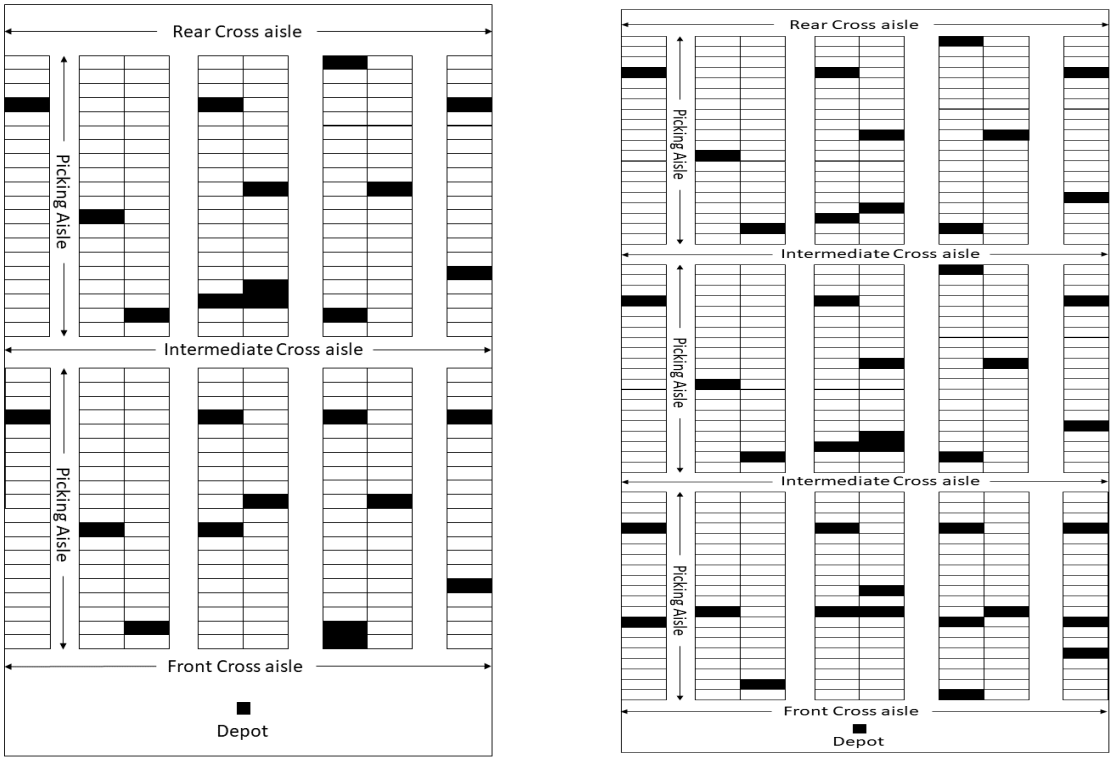
These results (Figure 5.3) confirm that the AMR speed affects the order picking

performance, but its influence is also dependent on the picker speed. Specifically, the findings suggest that it is beneficial to have an AMR that travels faster than the picker but also exhibits the law of diminishing marginal utility. In other words, when the picker speed is fixed, the percentage improvement derived from each unit increase in AMR speed decreases. Besides, the results also suggest an interaction between robot-human team composition and AMR speed. If the AMR travels faster than the picker, then it may be beneficial to employ fewer AMRs than pickers.

5.9 Impact of Multi-Block Warehouse Layout

The analysis in the previous subsections considered a single-block warehouse layout since it is most commonly studied in the literature, including the recent research on CHR-OPS (Žulj et al., 2021).In practice, warehouses dealing with a large volume and variety of items typically adopt a multi-block layout with intermediate cross-aisles. Nevertheless, the proposed approach applies to any warehouse layout because it requires only the distance between the item locations within the warehouse, and this information can be easily calculated for any layout. This section considers two-block and three-block warehouse layouts and uses the proposed RSA-ANS approach to solve instances involving 50 and 100 items. As shown in Figure 5.4, an intermediate cross-aisle separates the blocks in multi-block warehouse layouts. Similar to the baseline setting discussed in Section 3.5, each block has 20 racks and 400 storage locations.

Besides, the dimensions of the rack and storage locations are set to be the same as a single-block layout. Therefore, a two-block layout has 40 racks (800 storage locations), and a three-block layout has 60 racks (1200 storage locations). All the remaining settings are set to the baseline value.



(a) (b)

Figure 5.4: Illustration of (a) two-block and (b) three-block warehouse layouts

Figures 5.5(a) and 5.5(b) compares the tardiness value pertaining to different warehouse layouts for 50 and 100-item instances, respectively. The impact of due-date tightness (γ) and human-robot composition on total tardiness is similar for the three layouts. In other words, irrespective of the warehouse layout, the tardiness values increase as the due-date tightness increases. Likewise, increasing the number of pickers

(slower resource) leads to the greatest improvement in tardiness. We can also observe that the tardiness value increases with the number of blocks. This is expected because the distance traveled by the AMRs and pickers is likely to increase as the warehouse size (number of blocks and cross aisles) increases. Thus, the results suggest that the impact of warehouse layout on tardiness is not negligible. Further, it also demonstrates the capability of the proposed solution approach to handle multi-block layouts.

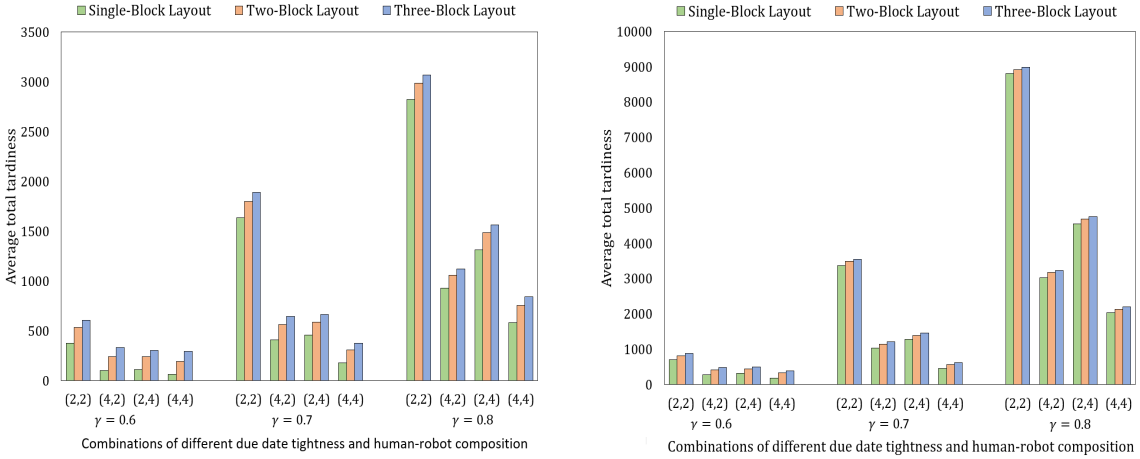


Figure 5.5: Impact of different warehouse layouts on total tardiness for instances with (a) 50 and (b) 100 items

5.10 Conclusions

In this chapter, we propose a restarted simulated annealing algorithm with adaptive neighborhood search (RSA-ANS) to deal with the computational intractability of the optimization model, which integrates an adaptive multi-neighborhood search

and fix-andoptimize restart technique with simulated annealing. Extensive numerical analysis showed that the RSA-ANS is able to achieve the optimal or near-optimal (within 0.2%) solution for all the small problem instances. Besides, the proposed RSA-ANS achieved superior solution quality for larger instances, and outperformed the variable neighborhood descent (VND) algorithm with an average improvement of about 13%. In addition, employing the proposed optimize-and-fix restart strategy yields about 7% lower tardiness, suggesting its capability to escape local optima. Our results also provided several practical insights on CHROBASRP, particularly the influence of AMR cart capacity, AMR speed and human–robot team composition.

Chapter 6

Collaborative Dynamic Order Picking System

6.1 Introduction and Motivation

The rapid increase in customer demands and expectations in the Business-to-Consumer (B2C) segment of e-commerce has put immense pressure to improve warehouse operations (Giannikas et al., 2017; Lu et al., 2016; De Koster et al., 2007). Consumer orders are increasingly becoming smaller, more frequent, and placed at any time (Lu et al., 2016; De Koster et al., 2007). The customers expect express deliveries during a time slot convenient for them. Further, they also demand late

cancellations and modifications to their orders, like adding or dropping an item and a change of delivery location or time (Giannikas et al., 2017; Lu et al., 2016). Hence warehouses have been pressurized to optimize all their operations and strive for high responsiveness while keeping costs low.

To remain competitive and differentiate services, e-commerce companies increasing allow for last minute orders and customizations and strive to improve responsiveness. (Lu et al., 2016; De Koster et al., 2007). Further, to achieve economies of scale, many companies have strategically opted to operate few but larger warehouses, increasing the order-picking time and making it challenging to fulfill this growing need for speed and flexibility (De Koster et al., 2007; Lu et al., 2016). Existing static approaches to order picking are not effective for today's warehouses, as these methods expect the information on customer orders to be readily available beforehand. While in reality, the customer orders dynamically appear as the shift progresses (Fibrianto & Hong, 2019; Giannikas et al., 2017). Thus, the order arrivals in OPP must be considered dynamic over the planning horizon (Fibrianto & Hong, 2019). In other words, there is a dire need for dynamic order-picking systems that can almost instantaneously adapt their pick lists and schedules upon any disruption like new orders, urgent picks, discrepancies, or late changes.

Increased customer orders and larger warehouses also underline the obstacle concerning aggravated labor requirements and their associated costs (Lee & Murray,

2019). Furthermore, the labor shortage has been a significant issue in logistics and distribution (Bhattacharjee et al., 2021). COVID-19 has only exacerbated the problem, as the post-pandemic era faces "great attrition," with over 19 million US workers quitting, resulting in increased labor competition and higher costs (De Smet et al., 2021). Finally, the pandemic also witnessed direct disruptions to numerous supply chains owing to acute labor shortages from government-enforced lockdowns and workers falling ill, creating a pressing need to make distribution centers more resilient in the future (Bhattacharjee et al., 2021). Many fulfillment centers seek to solve the above labor shortage issues by automating their operations.

In the order picking system, around 50% of an order-pickers time is spent traversing between different picking points, with only 15% of their time dedicated to the actual picking process. Further, this prolonged item transportation time is also associated with worker health issues (Tompkins et al., 2010). Thus, by employing AMRs to perform this traversing task, labor availability can be increased to enable effective order processing and scalability, along with improving worker health conditions. On the contrary, the actual picking process requires more sophistication and flexibility, which is better handled by human pickers (Lee & Murray, 2019), ensuring employment for the workers. Thus, a collaborative human-robot order-picking system (CHR-OPS) capable of handling dynamic orders can be a promising solution to the operational and socio-economic challenges discussed.

Motivated by the challenges posed by e-commerce and the advantages offered by AMRs, this chapter aims to develop an interventionist approach to CHR-OPS that can operate efficiently in presence of dynamic customer orders. Interventionist strategy aims to improve the responsiveness and completion time of the CHR-OPS by intervening and amending the earlier order picking decisions as and when new orders appear.

6.2 Methodology

In this section, we describe the proposed CHR-OPP in detail with respect to the introduced interventionist and select-interventionist algorithms.

6.2.1 Problem Description

The OPS considered here is based on a block warehouse layout consisting of picking aisles and cross aisles. The parrel aisles with racks on either side to store SKUs are called the picking aisles. Their storage bins can only be accessed by human pickers (i.e., AMRs cannot retrieve items from the bins). The pathways perpendicular to the picking aisles that provide passage for the pickers and AMRs to travel between the picking aisles are called the cross-aisles. The cross-aisles do not contain storage bins to hold the SKUs. However, the racks of the picking aisle consist of multiple shelves

containing storage bins to hold SKUs. Each position (i.e., left to right) in the shelf is considered to contain multiple items of the same SKU. Thus, if there are two orders for the same item, the SKU and location will be the same. The picking position of an item of any SKU can be represented by its aisle, rack, shelf, and position numbers. For instance, A02-R04-S01-P03 denotes the second aisle, fourth rack, first shelf, and third position on that level from left. There are no storage locations along the path of both the picking aisles and cross aisles, but these paths are accessible to both human pickers and AMRs for travel.

In this study, we assume the presence of initial orders before the picking shift begins. As the picking shift proceeds, new orders may arrive every second, following a Poisson distribution. If an order consists of multiple items, each item is considered as a separate order. The picking process involves retrieving items from the racks and delivering them to a single depot. The depot is denoted as location 0, and the first location of the first rack is denoted as location 1, followed by 2, 3, 4, and so on. For example, $d_{0,108}$ means the distance from the depot to location 108. Since the picking aisles are vertical and cross aisles are horizontal, the Manhattan distance is used to calculate the distance between any two locations.

In the order picking process, a group of AMRs (i.e. transport robots) $r \in \mathcal{R}$ and human pickers $k \in \mathcal{K}$ work together in a collaborative manner. Each AMR is responsible for transporting assigned items to the depot using a cart with a capacity

limit κ_r. An AMR can make multiple tours, counted as δ, and transport a subset of a collection of items in each tour. On the other hand, human workers perform the pick-and-place operation, which takes η_i^L time units to retrieve an item from the rack and load it onto the AMR's cart. At the depot, it takes η_i^U time units to unload each item from the AMR's cart. While travelling in collaborative scenarios, human pickers have a speed of v^K, and AMRs have a speed of v^R. In the human-only scenario, the speed of the human is $0.6 * v^K$, since they need to carry the cart. The time needed for a human picker k to move from one location i' to another location i is denoted as $\tau_{i',i}^K$, which is equal to $d_{i',i}/v^K$, same as that of the AMRs.

At the beginning of the shift, the AMRs will wait at the depot and the human pickers will wait at their designated zones. Each AMR will be assigned a mission list $\mathcal{M}_r$, containing the list of items along with their picking locations that the AMR has to visit to collect the assigned items. Further, depending on whether the AMRs are following an interventionist or non-interventionist strategy, their mission list can or cannot be updated during the tour after departure. Once an AMR starts moving towards the first/ next item on the mission list, a picker will be assigned to that item. The picker may potentially be working on another mission, but once they finish their task, they will start moving towards that item's location. Once the picker and the AMR meet at the pick location, the human picker will retrieve the item from the shelf and load it onto the AMR's cart. The AMR will then update the mission list and start moving to collect the next item on the mission list.

In a distribution warehouse, it is typical for each ordered item to have a due date to ensure timely delivery to external customers. The degree to which the due date is met is dependent on the successful resolution of key subproblems (Lee and Murray, 2019). If an order is fulfilled later than its due date, penalties may be incurred, which are typically based on the duration of the delay, also known as tardiness (Scholz et al, 2017). Tardiness refers to the delay or lateness of an operation or task beyond its scheduled or expected completion time. In other words, it is defined as the non-negative difference between the order completion time and its due date. It can cause disruptions to production schedules, lead to missed deadlines, and affect overall productivity and customer satisfaction. The average tardiness is calculated as the sum of the tardiness for all ordered items, divided by the total number of orders.

6.2.2 Notations

This section provides the notations used to describe the problem. These notations will be utilized to present the proposed interventionist and select-interventionist algorithms in the upcoming sections.

Indices and Sets

i	all the items stored in the warehouse
t	time in second

$r \in \mathcal{R}$	Set of robots
$k \in \mathcal{K}$	Set of pickers
$\mathcal{O}_t$	Set of items that will be ordered at time t
$\mathcal{P}$	Set of items in the pending list
$\mathcal{P}_z$	Set of all items in the pending list of zone z
$\mathcal{Y}_r$	Set of items carried by robot r
$\mathcal{M}_r$	Mission list for robot r
$\delta = 0$	Tour counter

Parameters

$d_{i',i}$	Distance between locations of item $i\prime$ and i
h	the length of the shift
v^K	Moving speed for human pickers
v^R	Moving speed for robots
$\tau^K_{i',i}$	Travel time from location of item i' to location of item i by human pickers
$\tau^R_{i',i}$	Travel time from location of item i' to location of item i by AMR
κ_r	Capacity of robot r in any tour, expressed as the number of bins which can be carried by the AMR
η^U_i	Time to retrieve (unload) item i from storage location
η^L_i	Time to place (load) item i on AMR's cart
X_i	The storage location of item i

X_0	The location of depot

Decision variables

d^K_{total}	The total moving distance of all human pickers
$A^K_{i,k}$	1 if picker k is assigned to pick-and-place item i; 0 otherwise
$A^R_{i,r}$	1 if item i is assigned to be transported by robot r; 0 otherwise
$X^R_{r,t}$	The location where the robot r is moving to or staying at time t
$X^K_{k,t}$	The location where the picker k is moving to or staying at time t
$B^R_{i,r}$	Collection begin time of item i by AMR r
$F^R_{i,r}$	Collection finish time of item i by AMR r
$S^R_{r,\delta}$	Start time of tour δ by robot r
$C^R_{r,\delta}$	completion time of tour δ by robot r

6.2.3 Definitions

In the literature review we have seen that most existing methods to dynamic order picking do not allow or restrict any modifications to the pick list once a robot or picker has started their tour. However, in a dynamic environment, the robots or picker can benefit from such modifications as it allows them to collect new orders which might only require small deviation from the remaining tour. Thus, we formally introduce intervention approach to CHR-OPP below.

- **Interventionist Approach to CHR-OPP:** A dynamic approach to CHR-OPP, wherein the current pick cycle can be altered by modifying the robot's pick list on receiving new orders. Such an intervention is made irrespective of the AMR or picker's current location, selected routing policy or the pick locations of the remaining items, but is restricted by the robot's carrying capacity.

However, not always will such an intervention improve the order picking performance. Especially in cases where the newly ordered item is located far away from the pending items, it may be a wise to pick the new item in the next tour. Hence there is a need to implement these interventions discerningly. Thus, we also propose and define a select- interventionist approach to order picking.

- **Rule-based Interventionist Approach to CHR-OPP:** A strategy to selectively implement the interventionist approach to CHR-OPP to ensure improvement in the overall efficiency. Here once the AMR has left the depot, its mission list is modified to include a newly placed order only if we know that picking that new item in the current tour will improve the overall performance.

6.2.4 Critical decisions regulating the interventionist approach to CHR-OPP

There are many critical decisions that impact the overall efficiency of the interventionist approach to CHR-OPP. Five of the important decisions are discussed below:

1. **Should zoning and picker/ AMR assignment be performed?** The zoning problem in order picking involves specifying different picking zones within a warehouse and assigning pickers to be stationed within these zones. Such an approach ensures that the human workers pick within a zone, and do not expend effort making long walks between zones. However, existing literature shows how zoning can both substantially improve picking efficiency, but also hinder effective operations in some cases, depending on the nature of the warehouse and the incoming customer orders (Azadeh et al., 2020). Thus, the proposed CHR-OPP algorithm has the provision for zoning and assigning each picker and AMR to a particular zone, as well as relaxing zoning by considering the entire warehouse as one zone.

2. **When should the pick list be updated?** Most of the existing literature on dynamic order processing does not allow the pick list to be updated once the AMR has left the depot. In such an approach, the new order is appended to the pending list. However, allowing new items to be added to the mission list

may help improve picking efficiency, but some other items may get delayed or added to the next tour. Finally, modifying the pick list with new orders will not always improve the overall picking efficiency. For instance, adding a new item to an AMR's mission list after it has collected all its items and has almost returned to the depot could be detrimental. Adding the new item to the next tour would prove beneficial in such scenarios.

3. **How is the AMR routing/ re-routing performed?** Most existing static routing algorithms cannot handle new incoming orders dynamically. Further, whenever the mission list is updated, the AMR has to be re-routed almost instantaneously and irrespective of its current (i.e., new starting) position, requiring computationally effective algorithms. Finally, prescribing simple and familiar intralogistics routes to the human pickers has been identified to improve picking performance. Thus we employ two of the most popular routing policies (i.e. S-shape and largest gap method) in intralogistics that satisfy the mentioned requirements. For the S-shaped routing policy, if the AMR needs an item in an aisle, it will go through the entire aisle and pick all other items needed from that aisle together. On the other hand, under the largest gap routing method, the robot will travel in the cross-aisle and enter a picking aisle only if an item to be picked is present within the mid-point of that aisle. Once it has picked all the items within that aisle's midpoint, it must exit the aisle from the same side (De Koster, et al., 1999).

4. **How is an item assigned to an AMR or Picker?** In the proposed system, whenever an AMR is moving towards the next item on the mission list, the closest available picker in that zone is assigned for carrying out the pick and place operation. In case there is no available picker, the system waits until any picker of that zone completes his missions, and then assigns that picker for the task. Similarly, whenever there is a new order, the new item to be picked is assigned to an AMR working in the same zone, or added to the pending list specific to that zone.

5. **What should be the AMR fleet characteristics?** In this study, by default we assume that both AMRs and human pickers travel at the same speed in CHR-OPP. The AMR capacity constraint determines the number items that can be picked on a single tour, by default this value is 20. However, we also vary these characteristics and study the impact of different speed ratios between AMR and picker, capacity constraint, and fleet size.

6.2.5 Picking policy design and CHR-OPP algorithms

This paper proposes two different interventionist algorithms that also leverage AMRs capable of collaborating with human workers for order picking.

6.2.5.1 Interventionist Picking Algorithm (CHR-IPA)

The assignment Algorithm 5 employed here ensures that a robot's current mission list can be intervened by the dynamic new orders, even if the robot is on a picking tour. The new item is assigned to the robot in the same zone as the item's pick location (lines 5-6). When more than one robot is available, the robot closest to the pick location is selected (line 8). The items which are already picked will be kept, but unpicked items may be delayed, or even assigned to future batches due to the addition of new orders ahead of them in the picking queue. Here, an AMR and a picker will travel separately along their respective routes, then meet at the designated item location and complete the loading process in collaboration. After each item is loaded on a AMR, the AMR will check the mission list for any updates that may have been made, including new orders. This allows for improved overall completion time and a more efficient scheduling process. With the arrival of new orders, all unpicked orders in the mission lists for the following tours, including the current one, will undergo a complete rescheduling process. Notably, we consider two variants for the proposed CHR-IPA.

Algorithm 5 Interventionist Assignment

1: **Inputs:** Problem parameters, item i^* to be assigned, current time t, locations $X^R_{r,t}$ $\forall r \in \mathcal{R}$ at time t, and mission lists $\mathcal{M}_r$ $\forall r \in \mathcal{R}$
2: **Output:** AMR assignment $A^R_{i^*,r'}$ for item i^*
3:
4: **Procedure:**
5: Find the zone Z_{i^*} holding item i^*
6: **if** r' is the only robot in Z_{i^*} **then**
7: assign $A^R_{i^*,r'} = 1$
8: **else**
9: assign it to the closest robot r' : $\min d_{X^R_{r',t},X_{i^*}}$ within the zone Z_{i^*}
10: **end if**

- CHR-IPA(D): The letter "D" indicates that when assigning a picker to meet with a robot at a picking location, the picker who has or will have the shortest distance to the item's location is selected.
- CHR-IPA(T): The letter "T" indicates that when assigning a picker to meet with a robot, the picker who will cause the least robot wait time at the picking location is selected.

The proposed interventionist CHR-OPP is discussed in Algorithm: 6. First, problem parameters and the selected routing strategy are provided as input (line 1). Then, the current time, tour counter, average walking distance for humans, and the locations of pickers and robots are initialized (lines 2-3). During the order-picking process, a new order is assigned to a robot based on the proposed interventionist assignment Algorithm 5 (lines 4-6). Following an order assignment, the items in a robot's mission list are sequenced using the routing strategy given as input (line 7). Robots that are not assigned any task are instructed to move to the depot and keep waiting, and their state variables are updated (lines 10-21). In the case of a human-only strategy, the

worker moving distance is also updated (lines 17-18). If a robot is at the depot and the mission list is not empty, the robot departs on its tour and visits the first pick location, and the events are updated (lines 24-28). A suitable picker is also instructed to move to that pick location based on the selected strategy (line 30). In the collaborative scenario, the assigned picker must complete his ongoing task before moving to that pick location (line 30). Once both the picker and robot have arrived, the timeline is updated, and the loading activity begins (lines 31-38). The total picker travel distance is updated according to the selected strategy (lines 40-43). After loading a picked item, the robot's capacity is checked to determine the next action (line 51). If the robot has the capacity, it continues to move to the next location on the pick list; else, it is instructed to go back to the depot and unload; (lines 51-54). Following this, the timeline and state variables are updated (lines 53-54).

Algorithm 6 Interventionist

Inputs: problem parameters, routing method selected
Outputs: $F^R_{i,r}, B^R_{i,r}, S^R_{r,\delta}, C^R_{r,\delta}, d^K_{total}$
Initialize: t, δ, d^K_{total}, $X^R_{r,0}$, $X^K_{k,0} = 0$

Procedure:

while $t \leq h$ or $\mathcal{M}_r \neq \emptyset\ \forall r$ or $\mathcal{Y}_r \neq \emptyset\ \forall r$ **do**

  if $|\mathcal{O}_t| > 0$ **then**
    Assign each item in $\mathcal{O}_t$ to a mission list $\mathcal{M}_r$ of a robot r by Algorithm 5
    Sequence each mission list $\mathcal{M}_r\ \forall r$ using the selected routing method
  end if

  for $r \in R$ **do**
    if $X^R_{r,t} = \emptyset$ **then**
      if $\mathcal{M}_r = \emptyset$ **then**

        if $X^R_{r,t^r_{max}} = X_0$ **then**
          $X^R_{r,t} = 0$
        else
          for t' in range int$(d_{X^R_{r,t^r_{max}},X_0}/v^{\mathcal{R}} + \eta^U)$ **do**
            $X^R_{r,t'} = X_0$, $C^R_{r,\delta} = t$, $\delta = \delta + 1$
          end for

          if human-only strategy **then**
            $d^K_{total} = d^K_{total} + d_{X^R_{r,t^r_{max}},X_0}$
          end if

          for $i' \in \mathcal{Y}_r$ **do**
            $F^R_{i',r} = t$
          end for
          $\mathcal{Y}_r = \emptyset$
        end if

      else
        if $X^R_{r,t^r_{max}} = X_0$ **then**
          $B^R_{i,r} = t$, $S^R_{r,\delta} = t$
        end if
        remove the first item $i_{[1]} \in \mathcal{M}_r$
        for t' in range int$(d_{X^R_{r,t},X_{i_{[1]}}}/v^{\mathcal{R}})$ **do** $X^R_{r,t'} = X_i$
        end for
        if collaborative strategy **then**
          t^k_{arrive} =max(t, t^k_{max})+int$(d_{X^K_{k,t^k_{max}},X_{i_{[1]}}}/v^{\mathcal{K}})$
          $t^r_{arrive} = t^r_{max}$+int$(d_{X^{\mathcal{R}}_{r,t^r_{max}},X_{i_{[1]}}}/v^{\mathcal{R}})$
          t_{arrive} =max$(t^k_{arrive}, t^r_{arrive})$
        else
          $t_{arrive} = t^r_{max}$+int$(d_{X^{\mathcal{R}}_{r,t^r_{max}},X_{i_{[1]}}}/v^{\mathcal{R}})$
        end if
        for $t^r_{max} < t' \leq t_{arrive} + \eta^L$ **do** $X^R_{r,t'} = X_{i_{[1]}}$
        end for
        if human-only strategy **then** $d^K_{total} = d^K_{total} + d_{X^R_{r,t^r_{max}},X_{i_{[1]}}}$
        else
          $d^K_{total} = d^K_{total} + d_{X^K_{k,t^k_{max}},X_{i_{[1]}}}$
        end if

Algorithm 7 Interventionist (continued)

55: **if** $t^k_{max} < t$ **then**
56: **for** $t^k_{max} < t' \leq t$ **do** $X^K_{k,t'} = X^K_{k,t^k_{max}}$
57: **end for**
58: **end if**
59: **for** $t^k_{max} < t' \leq t_{arrive}$ **do** $X^{\mathcal{K}}_{k,t'} = X_{i_{[1]}}$
60: **end for**
61: append $i_{[1]}$ to $\mathcal{Y}_r$
62: **if** $|\mathcal{Y}_r| \geq \kappa_r$ **then**
63: **for** $t^r_{max} < t' \leq t^r_{max}+\text{int}(d_{X^R_{r,i_{[1]}},X_0}/v^{\mathcal{R}})+\eta^U$ **do**
64: $X^R_{r,t'} = X_0$, $C^R_{r,\delta} = t$, $\delta = \delta + 1$
65: **end for**
66: **if** human-only strategy **then** $d^K_{total} = d^K_{total} + d_{X^R_{r,t^r_{max}},X_0}$
67: **end if**
68: **for** $i' \in \mathcal{Y}_r$ **do** $F^R_{i',r} = t$
69: **end for**
70: $\mathcal{Y}_r = \emptyset$
71: **end if**
72: **end if**
73: **end if**
74: **end for**
75: **end while**

6.2.5.2 Rule based Interventionist Picking Algorithm (CHR-RIPA)

The assignment Algorithm 8 employed here ensures that a robot's current mission list can be intervened by the dynamic new orders, even if the robot is on a picking tour, but only if the addition of the new item will not increase the average travel distance per item in the tour. The new item can be assigned directly to a robot's current mission list following an interventionist approach or can be assigned to the pending list causing no interventions. This decision depends on the calculated expected worker travel distance to pick up the ordered items (lines 2-11). If assigning the new item to a robot's mission list reduces the average distance per pick, the intervention is made, else the item is appended to the pending list (lines 12-20).

Algorithm 8 Rule-based Interventionist Assignment

1: **Inputs:** Problem parameters, item i^* to be assigned, current time t, locations $X^R_{r,t}$ $\forall r \in \mathcal{R}$ at time t, and mission lists $\mathcal{M}_r$ $\forall r \in \mathcal{R}$
2: **Output:** AMR assignment $A^R_{i^*,r'}$ for item i^*
3: **Defined Functions**
4: (a) Avg_Dist(S): returns the average distance for picking items in a given sequence S.
5: (b) Sequence(I): returns the picking sequence for a set of I items using S-shape strategy.
6:
7: **Procedure:**
8: Find the zone Z_{i^*} holding item i^*
9: **if** there are free robots r_{free} waiting at depot with $M_r = \emptyset$ **then**
10: assign i^* to any free robot r_{free}, $A^R_{i^*,r_{free}} = 1$
11: **else**
12: **if** Avg_Dist(Sequence($i^* + \mathcal{M}_r$)) $\leq$ Avg_Dist(Sequence($\mathcal{M}_r$)), for any $r \in \mathcal{R}$ **then**
13: append i^* to $\mathcal{M}_r$, where Avg_Dist(Sequence($i^* + \mathcal{M}_r$)) is smallest
14: **else**
15: append i^* to the pending list $\mathcal{P}_z$ of zone Z_{i^*}
16: **end if**
17: **end if**

This approach ensures that modifying the pick list does not significantly impact the overall travel distance, thus improving the efficiency of the picking process. Once a new item is added to the current batch, the existing items in the batch may be re-sequenced to optimize the picking process. If there are new orders added to the pending list, the following batches after the current one may be potentially re-batched. Similar to the Interventionist approach, the items which are already picked will be kept, but unpicked items may be delayed, or even assigned to future batches due to the addition of new orders ahead of them in the picking queue. A robot and a picker will travel separately along their respective routes, then meet at the designated item location and complete the loading process in collaboration.

The proposed rule-based interventionist CHR-OPP is discussed in Algorithm: 9. While the required inputs and initialization steps are similar, Algorithm 8 is leveraged instead of Algorithm 6 to assign new orders to a robot's mission list or a zone's pending list (line 6). Any robot with a completed mission list must return to the depot, unload

all the picked items, and update the tour completion time (lines 10-14). If the mission list is not empty (line 31), the robot is instructed to move to the next pick location. The steps for collaborating with a selected picker and updating time line and state variables are similar to the previous interventionist approach discussed.

Algorithm 9 Ruled Interventionist

Inputs: problem parameters, routing method
$t = 0$, $\delta = 0$, $d^K_{total} = 0$
initialize $X^R_{r,0}$, $X^K_{k,0}$
while $t \leq h$ or $\mathcal{M}_r \neq \emptyset\ \forall r$ or $\mathcal{Y}_r \neq \emptyset\ \forall r$ **do**
  if $|\mathcal{O}_t| > 0$ **then**
    assign all items in $\mathcal{O}_t$ to $\mathcal{M}_r$ and $\mathcal{P}_z$ by Algorithm 2
    sequence $\mathcal{M}_r\ \forall r$ by routing method
  end if
  for $r \in R$ **do**
    if $X^R_{r,t} = \emptyset$ **then**
      if $\mathcal{M}_r = \emptyset$ **then**
        if $X^R_{r,t^r_{max}} \neq X_0$ **then**
          for t' in range $\text{int}(d_{X^R_{r,t^r_{max}},X_0}/v^{\mathcal{R}} + \eta^U)$ **do**
            $X^R_{r,t'} = X_0$, $C^R_{r,\delta} = t$, $\delta = \delta + 1$
          end for
          for $i' \in \mathcal{Y}_r$ **do** $F^R_{i',r} = t$
          end for
          $\mathcal{Y}_r = \emptyset$
        else if $t^r_{max} \leq t$ **then**
          if $\mathcal{P}_z = \emptyset$ **then** $X^R_{r,t} = 0$
          else
            for $i' \in \mathcal{P}$ **do**
              if $|\mathcal{M}_r| \leq \kappa$ **then** remove i' from $\mathcal{P}_z$ and append to $\mathcal{M}_r$
              end if
            end for
            sequence $\mathcal{M}_r$ according to selected rule
            $B^R_{i,r} = t$, $S^R_{r,\delta} = t$
          end if
        end if
      end if
      if $\mathcal{M}_r \neq \emptyset$ **then**
        remove the first item $i_{[1]} \in \mathcal{M}_r$
        for t' in range $\text{int}(d_{X^R_{r,t},X_i}/v^{\mathcal{R}})$ **do** $X^R_{r,t'} = X_{i_{[1]}}$
        end for
        select k that minimize $d_{X^K_{k,t^k_{max}},X_{i_{[1]}}}$
        $A^K_{i_{[1]},k} = 1$
        $t^k_{arrive} = \max(t, t^k_{max}) + \text{int}(d_{X^K_{k,t^k_{max}},X_{i_{[1]}}}/v^{\mathcal{K}})$
        $t^r_{arrive} = t^r_{max} + \text{int}(d_{X^{\mathcal{R}}_{r,t^r_{max}},X_{i_{[1]}}}/v^{\mathcal{R}})$
        $t_{arrive} = \max(t^k_{arrive}, t^r_{arrive})$
        for $t^r_{max} < t' \leq t_{arrive} + \eta^L$ **do** $X^R_{r,t'} = X_i$
        end for
        $d^K_{total} = d^K_{total} + d_{X^K_{k,t^k_{max}},X_i}$

Algorithm 10 Rule-based Interventionist (continued)

if $t^k_{max} < t$ **then**
for $t^k_{max} < t' \leq t$ **do** $X^K_{k,t'} = X^K_{k,t^k_{max}}$
end for
end if
for $t^k_{max} < t' \leq t_{arrive}$ **do** $X^{\mathcal{K}}_{k,t'} = X_i$
end for
append i to $\mathcal{Y}_r$
if $|\mathcal{Y}_r| \geq \kappa_r$ **then**
for $t^r_{max} < t' \leq t^r_{max} + \text{int}(d_{X^R_{r,i},X_0}/v^{\mathcal{R}}) + \eta^U$ **do**
$X^R_{r,t'} = X_0$, $C^R_{r,\delta} = t$, $\delta = \delta + 1$
end for
if human-only strategy **then** $d^K_{total} = d^K_{total} + d_{X^R_{r,t^r_{max}},X_0}$
end if
for $i' \in \mathcal{Y}_r$ **do** $F^R_{i',r} = t$
end for
$\mathcal{Y}_r = \emptyset$
end if
end if
end if
end for
end while
return $B^R_{i,r}, S^R_{r,\delta}, C^R_{r,\delta}, F^R_{i,r}, d^K_{total}$

6.2.5.3 Performance Measurement

The performance of the system is evaluated based on several critical metrics including the total travel distance of humans, the completion time of orders, and the tardiness in full filling the orders. Only the travel distance of the human pickers is considered as the maintenance cost from AMR travel is not significant. Further, the AMRs also possess sufficient charge as the shift only last for 8 hours. Thus, average value for travel distance for each picker is calculated. Completion time represents the average time required to complete the picking process for each item within the shift, taking into account all dynamically arriving orders. Each item has a due date, and these dates are different for each order, and we assume that they follow a uniform

distribution. If an item is picked before its due date, then its tardiness is zero. However, if the item is not picked by the due date, we calculate the difference between the pick time and the due date to obtain the tardiness value. Finally, we compute the average tardiness for each item. Thus, the performance of the proposed algorithms are compared against that of the benchmark methods.

6.2.6 Benchmark Approaches

This study benchmarks the proposed algorithms against the Human only Interventionist Picking Algorithm (to understand the gains from employing collaborative robots) and CHR-OPP with No Interventions (to study the advantages of using an interventionist strategy).

6.2.6.1 Collaborative with No Interventions (CHR-NPA)

Here, the dynamic new order will not impact robots' mission lists. Instead, the new items will be added to pending lists according to which zone it belongs to. To give utmost priority to the current orders and to avoid delaying the completion time for the entire batch, it is preferable to add new orders only to the upcoming tours rather than the current tour. While the robot goes back to the depot, the new items can be added to the mission list along with old existing items in the pending list. A robot and a picker will travel separately along their respective routes, then meet

at the designated item location and complete the loading process in collaboration. Notably, we consider two variants for the proposed CHR-NPA.

- CHR-NPA(C): The letter "C" indicates that the robot will remain at the depot until there are sufficient new orders to fill its capacity or cart in a trip.
- CHR-IPA(5): The number "5" indicates that the robot will remain at the depot until there are 5 or more orders to be picked by that robot

The studied no interventionist CHR-OPP is discussed in Algorithm: 11. The primary distinction is that the newly arrived orders will be allocated to the pending list of a zone instead of directly being added to the mission list of a robot (line 6). Like previous strategies, the robot should return to the depot if it is free and the mission list is empty (lines 12-17). Only when the robots are at the depot, items are moved from the pending list to their mission lists. Following this, the items within a robot's mission list are sequenced using the routing strategy, and the robot departs on the pick tour and cannot be intervened by new incoming orders (lines 19-27). The process for collaboration, picking, and unloading remains the same.

Algorithm 11 No Interventionist

Inputs: problem parameters, routing method selected
Outputs: $B^R_{i,r}, F^R_{i,r}, S^R_{r,\delta}, C^R_{r,\delta}, d^K_{total}$
Initialize: t, δ, d^K_{total}, $X^R_{r,0}$, $X^K_{k,0} = 0$

Procedure:
while $t \leq h$ or $\mathcal{M}_r \neq \emptyset\ \forall r$ or $\mathcal{Y}_r \neq \emptyset\ \forall r$ **do**

  if $|\mathcal{O}_t| > 0$ **then**
    assign each item i in $\mathcal{O}_t$ to the pending list $\mathcal{P}_z$ of zone z with pick location X_i
  end if

  for $r \in R$ **do**
    if $X^R_{r,t} = \emptyset$ **then**
      if $\mathcal{M}_r = \emptyset$ **then**
        if $X^R_{r,t^r_{max}} \neq X_0$ **then**
          for t' in range $\text{int}(d_{X^R_{r,t^r_{max}}, X_0}/v^{\mathcal{R}} + \eta^U)$ **do**
            $X^R_{r,t'} = X_0$, $C^R_{r,\delta} = t$, $\delta = \delta + 1$
          end for
          for $i' \in \mathcal{Y}_r$ **do** $F^R_{i',r} = t$
          end for
          $\mathcal{Y}_r = \emptyset$
        else
          if $\mathcal{P}_z$=$\emptyset$ or fail to meet the conditions required for departure **then**
            $X^R_{r,t} = 0$
          else
            for $i' \in \mathcal{P}_z$ **do**
              if $|\mathcal{M}_r| \leq \kappa$ **then** remove i' from $\mathcal{P}_z$ and append to $\mathcal{M}_r$
              end if
            end for
            sequence $\mathcal{M}_r$ according to selected rule
            $B^R_{i,r} = t$, $S^R_{r,\delta} = t$
          end if
        end if
      end if

      if $\mathcal{M}_r \neq \emptyset$ **then**
        remove the first item $i_{[1]} \in \mathcal{M}_r$
        select k that minimize $d_{X^K_{k,t^k_{max}}, X_{i_{[1]}}}$
        $A^K_{i_{[1]},k} = 1$
        t^k_{arrive} =$\max(t, t^k_{max})$+$\text{int}(d_{X^K_{k,t^k_{max}}, X_{i_{[1]}}}/v^{\mathcal{K}})$
        $t^r_{arrive} = t^r_{max}$+$\text{int}(d_{X^{\mathcal{R}}_{r,t^r_{max}}, X_{i_{[1]}}}/v^{\mathcal{R}})$
        t_{arrive} =$\max(t^k_{arrive}, t^r_{arrive})$
        for $t^r_{max} < t' \leq t_{arrive} + \eta^L$ **do** $X^R_{r,t'} = X_{i_{[1]}}$
        end for
        $d^K_{total} = d^K_{total} + d_{X^K_{k,t^k_{max}}, X_{i_{[1]}}}$

Algorithm 12 No Interventionist (continued)

if $t^k_{max} < t$ **then**
for $t^k_{max} < t' \leq t$ **do** $X^K_{k,t'} = X^K_{k,t^k_{max}}$
end for
end if
for $t^k_{max} < t' \leq t_{arrive}$ **do** $X^{\mathcal{K}}_{k,t'} = X_{i_{[1]}}$
end for
append $i_{[1]}$ to $\mathcal{Y}_r$
if $|\mathcal{Y}_r| \geq \kappa_r$ **then**
for $t^r_{max} < t' \leq t^r_{max} + \text{int}(d_{X^R_{r,i_{[1]}}, X_0}/v^{\mathcal{R}}) + \eta^U$ **do**
$X^R_{r,t'} = X_0$, $C^R_{r,\delta} = t$, $\delta = \delta + 1$
end for
for $i' \in \mathcal{Y}_r$ **do** $F^R_{i',r} = t$
end for
$\mathcal{Y}_r = \emptyset$
end if
end if
end if
end for
end while

6.2.6.2 Human-only Interventionist

Here, a picker's current mission list can be interventioned by the dynamic new orders, even if the picker is on a picking tour. The items which are already picked will be kept, but unpicked items may be delayed, or even assigned to future batches due to the addition of new orders ahead of them in the picking queue. The picker will manually complete the entire picking process while using a cart.

6.3 Results

In this section, we examine the performance of the proposed and benchmark models under multiple settings. In particular, we perform extensive experiments to (i)

Evaluate the proposed models, namely CHR-IPA(D), CHR-IPA(T), and CHR-RIPA, against benchmark systems, namely CHR-NPA(C), CHR-NPA(5), and H-IPA (ii) Compare performances using holistic measures like travel distance, completion time, and tardiness, (ii) Gain insights on impact of order arrival rates, sequencing and routing rules, robot capacity, and warehouse zoning. All the algorithms considered in this chapter are coded using Python 3.10 and executed on a computer equipped with an Intel Core i5 8th generation processor and 64GB of RAM.

6.3.1 Generation of Test Instances

Similar to most prior works on order picking (Srinivas & Yu, 2022; Lee & Murray, 2019), we consider a single-block rectangular warehouse with 10 vertical picking aisles and 2 horizontal cross-aisles at the top and bottom. The picking aisles have shelves on both sides, with each shelf having 20 locations. Therefore, there are a total of 10 aisles x 2 sides x 20 shelves = 400 pick locations in the warehouse. The depot is located at the midpoint of the bottom cross aisle. The available space in the warehouse is divided into two equal zones. This study focuses on a dynamic online OPS where new orders can arrive every second of the 8 hours shift considered. At the start of the shift, we assume there will be a backlog of 20 items waiting to be picked. New orders will follow a Poisson distribution throughout the remainder of the shift. We consider three different arrival rates (orders/second) - 0.01, 0.03, and 0.05. For the

baseline analysis, we consider two AMRs and two pickers in each zone. Besides, each AMR or cart can carry up to 20 items.

Once an item is added to the mission list, the items are sequenced, and the robot is routed using either S-shape or largest gap method. We set the speed of the pickers and the robots to be 1m/s (Srinivas & Yu, 2022; Lee & Murray, 2019). For the human-only strategy, we consider a speed of 0.6 m/s as the operator must handle a cart to collect items (Srinivas & Yu, 2022; Lee & Murray, 2019) . The loading time for the picker (i.e., time taken to pick-and-place item from storage to AMR) is considered to be 5 seconds per item, and the unloading time at depot is 10 seconds per item (Srinivas & Yu, 2022; Lee & Murray, 2019). The loading and unloading times are assumed to be the same in manual picking as well. Each item to be picked corresponds to one location in the warehouse. Two methods are used to pair a robot with a picker for collaboration. The first approach is based on selecting the shortest walking distance. When a robot begins moving towards an item, the system will search for all available pickers in the same zone and choose the one whose final location is closest to the item's location. For the second approach, the system will compare all available pickers and select the one that causes least robot waiting time. This calculation considers the picker's current state, future location, distance to the item, and walking speed. Finally, the robots will need to return to the depot to unload the items while human pickers wait in their designated zones for next picking instruction. The proposed strategies are benchmarked and evaluated against Human-

only Interventionist Picking Algorithm (H-IPA), as well as the Collaborative Human-Robot order picking with no intervention (CHR-NPA). Each experiment was run for 30 iterations, and the reported results are the average values obtained across these 30 iterations.

The outcomes of the experiments are presented below. First, the performance of the proposed systems, along with that of the benchmark methods, in the default setting is given. The performance is measured using the following metrics: (1) average distance, (2) average completion time, (3) average tardiness, and (4) tardy rate. The average completion time measures the mean time (in seconds per item) needed to process/pick an order. Average tardiness measures the mean duration by which a picked item exceeds its expected due time (in seconds per item). The tardy rate measures the percentage of orders that reach the depot after the due time (i.e., have positive tardiness). Following the experiments in the default setting, five potentially important factors impacting the performance of the CHR-OPS are identified, their values are varied, and the results from further experiments are reported. The varied factors include (1) order arrival rate, (2) sequencing and routing rule, (3) capacity of the robots, (4) number of zones in the warehouse, and the (5) presence of heterogeneous order arrival rates (both through time and across zones).

6.3.2 Results for Baseline Setting

Table 6.4 compares the performance of the proposed systems CHR-IPA(D), CHR-IPA(T), and CHR-RIPA against the benchmark strategies CHR-NPA(C), CHR-NPA(5), and H-IPA in the mentioned default setting. In terms of completion time, CHR-NPA(C) takes the longest, about 5 to 22 times more than that of CHR-IPAs. The performance gap is especially large when the order arrival rate is low. The completion time of CHR-NPA(5), which has to wait for five orders (as opposed to the cart capacity in CHR-NPA (C)) before leaving the depot, performs better than CHR-NPA (C). However, it still performs relatively poor. For low-medium order arrival rates (i.e., 0.01 and 0.03 lambda values), the proposed CHR-RIPA provides the quickest order picking. On the other hand, the proposed CHR-IPAs provide better performance when the order arrival rates are higher (i.e., 0.05), with CHR-IPA(T) performing slightly better than CHR-IPA(D).

We observe a positive correlation between average completion time and average tardiness, and majority of the reported trends for completion time hold true for tardiness as well. This can be attributed to the reasoning that quick order-picking times are critical to avoid delays. On contrary, in terms of least distance traveled by a picker, CHR-NPA(C) perform the best. The humans in this system walk only about half as much as the pickers in CHR-IPAs. Here, the robots leave the depot only after they are assigned to their fullest capacity. Thus, the robot's route is dense

with pick locations, demanding less travel per item from the pickers. Further, since interventions are not allowed, there are no last-minute deviations (e.g., an intervention instructing the robot to pick a far away item when the robot has almost returned to the depot).

The second-best performance is obtained by CHR-RIPA when there is higher order arrival rate (i.e. 0.05) and CHR-NPA(5) when the order arrivals rates are lower (i.e. 0.01 or 0.03). As expected, the H-IPA approach results in the largest distance traveled, about 6 to 27 times that of the other approaches.

Table 6.4: Performance benchmarking in baseline setting

λ	Model	Distance(m)		Completion Time(s)		Tardiness(s)		Tardy Rate	
		μ	σ	μ	σ	μ	σ	μ	σ
0.01	H-IPA	70.98	1.79	302.7	16.11	11.73	5.61	0.93	0.02
	CHR-IPA(D)	5.35	0.22	105.33	5.53	2.06	1.41	0.98	0.01
	CHR-IPA(T)	5.45	0.23	103.07	3.49	1.72	1.17	0.98	0.01
	CHR-NPA(C)	2.62	0.07	2334.33	112.96	1474.89	117.11	0.16	0.03
	CHR-NPA(5)	5.06	0.17	603.57	23.57	123.79	18.01	0.7	0.03
	CHR-RIPA	5.3	0.2	101.57	3.62	3.3	1.68	0.98	0.01
0.03	H-IPA	44.76	1.13	386.17	14.48	23.85	4.95	0.88	0.01
	CHR-IPA(D)	5.37	0.13	132.5	5.08	2.87	0.97	0.96	0.01
	CHR-IPA(T)	5.49	0.13	129.43	3.3	2.46	0.65	0.96	0.01
	CHR-NPA(C)	2.65	0.04	1163.07	22.85	399.72	24.42	0.38	0.02
	CHR-NPA(5)	5.11	0.12	362.33	5.64	30.93	3.18	0.83	0.01
	CHR-RIPA	5.31	0.11	113.23	5.04	2.86	0.86	0.97	0.01
0.05	H-IPA	32.64	0.78	523.4	24.98	65.02	11.22	0.8	0.01
	CHR-IPA(D)	5.29	0.09	185	9.91	7.77	1.72	0.93	0.01
	CHR-IPA(T)	5.39	0.09	173.83	6.46	5.98	0.89	0.94	0.01
	CHR-NPA(C)	2.67	0.03	917.67	14.28	228.08	12.35	0.51	0.02
	CHR-NPA(5)	5.13	0.1	338.47	8.11	23.83	2.76	0.85	0.01
	CHR-RIPA	4.55	0.37	312.93	141.89	42.23	52.05	0.85	0.08

6.3.3 Impact of Routing Strategy

In this subsection, we compare the two different sequencing and routing strategy (1) the S-shape method and (2) the largest gap method. Table 6.5 shows that the proposed CHR-IPA(D), CHR-IPA(T), and CHR-RIPA achieve substantially better performance in majority of the scenarios across both routing strategies. Further, we can observe that both routing strategies perform comparably in most cases, with the S-shape method performing slightly better. For instance, in CHR-IPA(T) with lambda 0.01, the largest gap method is 1.1%, 3.5%, and 35%, inferior to the S-shape in terms of distance traveled, completion time, and tardiness, respectively. Notably, the percentage of tardiness difference is high due to the low tardiness values compared, which are around 2 seconds. Nevertheless, as the order arrival rate increases, the difference in performance between the two routing strategies becomes slightly pronounced. For instance, the difference in completion time between the routing strategies for CHR-IPA(D) is only 1.61% (i.e., 105.33s for S-shape to 107.03s for largest gap) when the lambda equals 0.01. However, this gap increases to 17.39% (i.e., 185.00s for the S-shape to 217.17s for the largest gap) when lambda equals 0.05. When the order arrival rate is low, the number of picking locations in an aisle is also low, making traversing an entire aisle a less efficient policy. Thus, we observe that the S-shape methods particularly performs better on high order arrival rate, when pick density per picking aisle is large.

Table 6.5: Impact of varying the routing strategy on picking performance

Routing Strategy	λ	Rule	Distance(m)		Completion Time(s)		Tardiness(s)		Tardy Rate	
			μ	σ	μ	σ	μ	σ	μ	σ
S-shape	0.01	H-IPA	70.98	1.79	302.7	16.11	11.73	5.61	0.93	0.02
		CHR-IPA(D)	5.35	0.22	105.33	5.53	2.06	1.41	0.98	0.01
		CHR-IPA(T)	5.45	0.23	103.07	3.49	1.72	1.17	0.98	0.01
		CHR-NPA(C)	2.62	0.07	2334.33	112.96	1474.89	117.11	0.16	0.03
		CHR-NPA(5)	5.06	0.17	603.57	23.57	123.79	18.01	0.7	0.03
		CHR-RIPA	5.3	0.2	101.57	3.62	3.3	1.68	0.98	0.01
	0.03	H-IPA	44.76	1.13	386.17	14.48	23.85	4.95	0.88	0.01
		CHR-IPA(D)	5.37	0.13	132.5	5.08	2.87	0.97	0.96	0.01
		CHR-IPA(T)	5.49	0.13	129.43	3.3	2.46	0.65	0.96	0.01
		CHR-NPA(C)	2.65	0.04	1163.07	22.85	399.72	24.42	0.38	0.02
		CHR-NPA(5)	5.11	0.12	362.33	5.64	30.93	3.18	0.83	0.01
		CHR-RIPA	5.31	0.11	113.23	5.04	2.86	0.86	0.97	0.01
	0.05	H-IPA	32.64	0.78	523.4	24.98	65.02	11.22	0.8	0.01
		CHR-IPA(D)	5.29	0.09	185	9.91	7.77	1.72	0.93	0.01
		CHR-IPA(T)	5.39	0.09	173.83	6.46	5.98	0.89	0.94	0.01
		CHR-NPA(C)	2.67	0.03	917.67	14.28	228.08	12.35	0.51	0.02
		CHR-NPA(5)	5.13	0.1	338.47	8.11	23.83	2.76	0.85	0.01
		CHR-RIPA	4.55	0.37	312.93	141.89	42.23	52.05	0.85	0.08
Largest Gap Method	0.01	H-IPA	71.31	1.65	314.97	18.09	13.62	6.7	0.92	0.02
		CHR-IPA(D)	5.39	0.21	107.03	4.02	2.25	1.65	0.98	0.01
		CHR-IPA(T)	5.51	0.22	106.63	5.04	2.12	1.48	0.98	0.01
		CHR-NPA(C)	2.84	0.07	2368.3	114.22	1504.18	118.74	0.15	0.03
		CHR-NPA(5)	5.06	0.17	603.57	23.57	123.79	18.01	0.7	0.03
		CHR-RIPA	5.36	0.2	106.3	4.43	4.42	2.09	0.98	0.01
	0.03	H-IPA	44.56	1.24	426	20.25	32.56	7.43	0.85	0.02
		CHR-IPA(D)	5.45	0.13	139.3	4.97	3.59	0.74	0.96	0.01
		CHR-IPA(T)	5.57	0.12	136.77	4.25	3.33	0.87	0.96	0.01
		CHR-NPA(C)	2.8	0.04	1194.23	23.11	419.21	24.73	0.36	0.02
		CHR-NPA(5)	5.11	0.12	362.33	5.64	30.93	3.18	0.83	0.01
		CHR-RIPA	5.36	0.11	121.53	11.38	4.7	2.28	0.97	0.01
	0.05	H-IPA	32.74	0.71	605.5	33.23	89.65	14.35	0.75	0.02
		CHR-IPA(D)	5.52	0.1	217.17	12.3	13.2	2.63	0.92	0.01
		CHR-IPA(T)	5.63	0.1	203.37	11.29	10.92	1.95	0.92	0.01
		CHR-NPA(C)	2.8	0.03	949.03	13.7	243.62	12.47	0.49	0.01
		CHR-NPA(5)	5.13	0.1	338.47	8.11	23.83	2.76	0.85	0.01
		CHR-RIPA	4.68	0.19	340.57	61.92	48.4	18.01	0.83	0.04

6.3.4 Impact of Robot Capacity

This sub-section investigates the impact of expanding the cart capacity of the robots on OPS performance. Capacity refers to the maximum number of items that a AMR or cart can carry in each tour. We study scenarios with 5, 10, 15, 20 item limits. Table 6.6 shows that as the capacity of the cart increases, the distance traveled by the picker reduces. This trend is observed for most systems, except for CHR-IPAs, where the robots travel more frequently to the depot to unload. Here the reported difference in performance is negligible, less than 0.2%. However, there is a trade-off when increasing the capacity. For instance, while the human travel distance decreases in a CHR-NPA(C) system (where the robots must wait at the depot to receive enough orders to meet the cart's capacity), the completion time significantly increases. However, this trend is less significant for CHR-IPAs and CHR-RIPA. Further, in a CHR-NPA(5) system where the robot has to wait at the depot for only 5 items regardless of the capacity, the impact on completion time is close to none.

6.3.5 Impact of Warehouse Zoning

Here we study the impact of zoning on the various performance measure. Pickers and robots are assigned to specific zones within the warehouse and are only responsible for handling items within their respective zones. All storage locations on a given shelf are assigned to the same zone, ensuring that a side of an entire aisle remains in

Table 6.6: Impact of varying the robot capacity on picking performance

Model	λ	Capacity	Distance		Completion Time		Tardiness		Tardy Rate	
			μ	σ	μ	σ	μ	σ	μ	σ
H-IPA	0.01	5	74.14	1.73	265.33	11.29	6.56	3.2	0.95	0.02
		10	73.73	1.69	263.67	9.3	5.44	2.43	0.95	0.02
		15	73.69	1.67	264.03	9.48	5.52	2.58	0.95	0.01
		20	70.98	1.79	302.7	16.11	11.73	5.61	0.93	0.02
	0.03	5	48.84	1.06	365.2	16.33	21.36	5.52	0.89	0.01
		10	47.24	1.23	350.17	12.22	15.94	3.38	0.9	0.01
		15	47.23	1.26	353	12.53	16.89	3.87	0.9	0.01
		20	44.76	1.13	386.17	14.48	23.85	4.95	0.88	0.01
	0.05	5	36.85	0.77	1267.8	331.42	741.05	308.94	0.69	0.03
		10	34.23	0.83	483.4	26.03	53.4	11.78	0.82	0.02
		15	33.95	0.81	483.87	19.32	50.32	7.51	0.82	0.01
		20	32.64	0.78	523.4	24.98	65.02	11.22	0.8	0.01
CHR-IPA(D)	0.01	5	5.35	0.22	102.3	4.85	1.58	0.91	0.98	0.01
		10	5.35	0.22	104.7	5.47	1.9	1.33	0.98	0.01
		15	5.35	0.22	105.33	5.53	2.06	1.41	0.98	0.01
		20	5.35	0.22	105.33	5.53	2.06	1.41	0.98	0.01
	0.03	5	5.38	0.12	129.43	3.23	2.44	0.71	0.97	0.01
		10	5.37	0.13	131.93	4.85	2.76	0.88	0.96	0.01
		15	5.37	0.13	132.47	5.07	2.86	0.98	0.96	0.01
		20	5.37	0.13	132.5	5.08	2.87	0.97	0.96	0.01
	0.05	5	5.27	0.09	178.33	8.82	6.78	1.89	0.94	0.01
		10	5.29	0.09	181.07	7.93	6.98	1.31	0.94	0.01
		15	5.3	0.09	184.5	10.2	7.65	1.68	0.94	0.01
		20	5.29	0.09	185	9.91	7.77	1.72	0.93	0.01
CHR-IPA(T)	0.01	5	5.44	0.22	100.47	3.41	1.33	0.74	0.98	0.01
		10	5.45	0.23	102.53	3.39	1.58	0.99	0.98	0.01
		15	5.45	0.23	103.07	3.49	1.72	1.17	0.98	0.01
		20	5.45	0.23	103.07	3.49	1.72	1.17	0.98	0.01
	0.03	5	5.5	0.11	127.2	2.45	2.19	0.59	0.97	0.01
		10	5.49	0.12	129.17	3.33	2.41	0.64	0.96	0.01
		15	5.49	0.13	129.4	3.27	2.45	0.65	0.96	0.01
		20	5.49	0.13	129.43	3.3	2.46	0.65	0.96	0.01
	0.05	5	5.4	0.1	166.83	5.05	5.15	0.98	0.95	0.01
		10	5.39	0.1	171.23	5.89	5.46	0.84	0.94	0.01
		15	5.39	0.09	173.47	6.5	5.9	0.84	0.94	0.01
		20	5.39	0.09	173.83	6.46	5.98	0.89	0.94	0.01
CHR-NPA(C)	0.01	5	5.16	0.19	596.73	22.23	124.26	17.7	0.7	0.02
		10	4.01	0.1	1179.4	59.22	478.35	51.3	0.41	0.03
		15	3.17	0.1	1764.1	81.87	956.22	80.83	0.25	0.03
		20	2.62	0.07	2334.33	112.96	1474.89	117.11	0.16	0.03
	0.03	5	5.21	0.1	353.47	5.75	28.93	2.93	0.84	0.01
		10	4.07	0.08	639.1	13.76	112.35	10.08	0.67	0.02
		15	3.22	0.06	907.8	20.2	238.13	19.12	0.51	0.03
		20	2.65	0.04	1163.07	22.85	399.72	24.42	0.38	0.02
	0.05	5	5.26	0.09	321.83	6.23	21.74	2.65	0.86	0.01
		10	4.12	0.05	529.57	6.09	68.06	4.97	0.73	0.01
		15	3.24	0.04	728.73	8.83	137.54	7.8	0.62	0.01
		20	2.67	0.03	917.67	14.28	228.08	12.35	0.51	0.02
CHR-NPA(5)	0.01	5	5.16	0.16	603.4	22.21	126.55	17.83	0.69	0.02
		10	5.08	0.14	604	21.92	126.28	17.27	0.69	0.02
		15	5.06	0.17	603.57	23.57	123.79	18.01	0.7	0.03
		20	5.06	0.17	603.57	23.57	123.79	18.01	0.7	0.03
	0.03	5	5.14	0.12	361.13	5.85	30.22	3.15	0.83	0.01
		10	5.11	0.11	363	6.03	31.21	3.56	0.83	0.01
		15	5.11	0.12	362.33	5.64	30.93	3.18	0.83	0.01
		20	5.11	0.12	362.33	5.64	30.93	3.18	0.83	0.01
	0.05	5	5.19	0.09	334.67	6.56	23.83	2.76	0.85	0.01
		10	5.14	0.1	338.57	7.03	23.79	2.22	0.85	0.01
		15	5.13	0.1	338.47	8.11	23.84	2.74	0.85	0.01
		20	5.13	0.1	338.47	8.11	23.83	2.76	0.85	0.01
CHR-RIPA	0.01	5	5.47	0.18	95.47	4.65	1.65	0.77	0.98	0.01
		10	5.39	0.19	95.97	4.99	1.96	1.37	0.98	0.01
		15	5.37	0.2	96.97	4.04	2.07	1.21	0.98	0.01
		20	5.37	0.2	96.97	4.04	2.07	1.21	0.98	0.01
	0.03	5	5.42	0.11	111.83	4.63	2.02	0.65	0.98	0.01
		10	5.39	0.1	112.3	5.13	2.21	0.88	0.98	0.01
		15	5.38	0.1	113.3	4.89	2.57	1.08	0.97	0.01
		20	5.39	0.1	113.77	5.06	2.73	1.2	0.97	0.01
	0.05	5	5.35	0.09	175.47	10.66	6.76	1.54	0.94	0.01
		10	5.25	0.07	180.93	10.6	7.89	1.77	0.94	0.01
		15	5.23	0.07	186.4	12.48	9.45	2.62	0.93	0.01
		20	5.22	0.07	187.33	13.29	10.05	3.41	0.93	0.01

the same zone. We study the picking problem under one, two, and four warehouse zone settings. A single zone system can be considered equivalent to a warehouse with no-zoning strategy. Observations from Table 6.7 indicate that the impact of increased zones is varied for different systems with respect to the distance traveled metric. For NPA strategies, it improves performance. However, for RIPA systems, the performance slightly reduces. For IPA algorithms, increasing the number of zones from 1 to 2 decreases travel distance, but a further increase in zones results in increased travel distance. With respect to completion time and tardiness, most IPA and RIPA-based systems significantly benefit from zoning. However, with NPA strategies, there is a substantial increase in completion time and tardiness on increasing the number of zones.

6.3.6 Impact of Heterogeneous Order Arrival Rate Across Zones

In the default setting, the order arrival rate is assumed to be homogeneous (i.e., similar) across different zones of a warehouse. However, in heterogeneous scenarios, the arrival rate varies across the zones. In this section we study the impact of storing similar items based on picking frequency in the same zone (resulting in heterogeneous order arrival rates across zones) as opposed to different zones (resulting in homogenous order arrival rates across zones). For the experiment, we assume that the order arrival rates for the two default zones differ by 20%. In terms of distance traveled

Table 6.7: Impact of varying the number of warehouse zones on picking performance

Method	λ	Zones	Distance		Completion Time		Tardiness		Tardy Rate	
			μ	σ	μ	σ	μ	σ	μ	σ
H-IPA	0.01	1	70.98	1.79	302.7	16.11	11.73	5.61	0.93	0.02
		2	73.69	1.67	264.03	9.48	5.52	2.58	0.95	0.01
		4	80.66	2.38	233.13	7.1	2.61	1.54	0.97	0.01
	0.03	1	44.76	1.13	386.17	14.48	23.85	4.95	0.88	0.01
		2	47.19	1.26	354	13.36	17.37	3.97	0.9	0.01
		4	52.52	1.44	302.67	9.56	9.13	1.82	0.93	0.01
	0.05	1	32.64	0.78	523.4	24.98	65.02	11.22	0.8	0.01
		2	33.87	0.81	485.57	23.06	51.5	9.07	0.82	0.02
		4	35.33	0.95	441.2	13.57	37.14	4.91	0.84	0.01
CHR-IPA(D)	0.01	1	5.34	0.21	121.3	7.88	4.8	3.66	0.97	0.01
		2	5.35	0.22	105.33	5.53	2.06	1.41	0.98	0.01
		4	5.8	0.21	92.1	2.91	0.67	0.47	0.99	0.01
	0.03	1	5.46	0.13	155.3	8.06	5.61	2.04	0.95	0.01
		2	5.37	0.13	132.5	5.08	2.87	0.97	0.96	0.01
		4	5.79	0.13	110.3	3.12	1.31	0.39	0.98	0.01
	0.05	1	5.36	0.11	223.3	13.79	13.88	3.34	0.91	0.01
		2	5.29	0.09	185	9.91	7.77	1.72	0.93	0.01
		4	5.7	0.1	147.53	5.54	3.98	0.8	0.96	0.01
CHR-IPA(T)	0.01	1	5.53	0.24	118.13	7.15	4.15	3.38	0.97	0.01
		2	5.45	0.23	103.07	3.49	1.72	1.17	0.98	0.01
		4	5.8	0.21	92.1	2.91	0.67	0.47	0.99	0.01
	0.03	1	5.57	0.14	144.5	4.29	3.68	0.94	0.96	0.01
		2	5.49	0.13	129.43	3.3	2.46	0.65	0.96	0.01
		4	5.79	0.13	110.3	3.12	1.31	0.39	0.98	0.01
	0.05	1	5.54	0.12	194.03	7.93	8.37	1.3	0.93	0.01
		2	5.39	0.09	173.83	6.46	5.98	0.89	0.94	0.01
		4	5.7	0.1	147.53	5.54	3.98	0.8	0.96	0.01
CHR-RIPA	0.01	1	5.33	0.19	118.23	4.37	8.24	3.59	0.97	0.01
		2	5.37	0.2	96.97	4.04	2.07	1.21	0.98	0.01
		4	5.88	0.22	94.07	3.39	0.85	0.53	0.98	0.01
	0.03	1	5.41	0.13	126.5	8.39	6.36	2.08	0.97	0.01
		2	5.39	0.1	113.77	5.06	2.73	1.2	0.97	0.01
		4	5.8	0.15	113.57	3.49	1.47	0.32	0.97	0
	0.05	1	5.17	0.08	293.8	35.75	36.1	11.78	0.87	0.02
		2	5.22	0.07	187.33	13.29	10.05	3.41	0.93	0.01
		4	5.61	0.1	147.47	3.79	3.41	0.66	0.96	0.01
CHR-NPA(5)	0.01	1	5.28	0.22	480.37	16.15	62.09	9.39	0.76	0.02
		2	5.06	0.17	603.57	23.57	123.79	18.01	0.7	0.03
		4	4.79	0.15	948.77	57.55	373.68	54.02	0.55	0.03
	0.03	1	5.34	0.12	361	7.5	28.7	3.22	0.83	0.01
		2	5.11	0.12	362.33	5.64	30.93	3.18	0.83	0.01
		4	4.76	0.1	467.7	9.92	67.66	6.34	0.77	0.01
	0.05	1	5.36	0.1	432.3	28.94	48.68	10.38	0.79	0.02
		2	5.13	0.1	338.47	8.11	23.83	2.76	0.85	0.01
		4	4.73	0.07	371.77	4.7	33.55	2.87	0.83	0.01
CHR-NPA(C)	0.01	1	3.92	0.11	1546.47	66.21	723.24	64.02	0.25	0.03
		2	2.62	0.07	2334.33	112.96	1474.89	117.11	0.16	0.03
		4	1.95	0.07	4027.33	228.13	3153.79	230.7	0.11	0.02
	0.03	1	4.01	0.08	988.7	13.88	261.02	16.45	0.47	0.02
		2	2.65	0.04	1163.07	22.85	399.72	24.42	0.38	0.02
		4	1.89	0.04	1709.9	49.96	891.59	49.53	0.23	0.02
	0.05	1	4.23	0.15	930.8	73.8	230.83	56.44	0.51	0.03
		2	2.67	0.03	917.67	14.28	228.08	12.35	0.51	0.02
		4	1.89	0.02	1224.97	23.21	458.29	20.81	0.36	0.01

by the picker in a collaborative system, Table 6.8 reports that homogenous zoning is beneficial for NPAs and IPA(T), and heterogenous zoning is beneficial for RIPA and IPA(D) systems. With respect to completion time and tardiness, homogenous zoning ensures the best performance in almost all scenarios.

6.3.7 Impact of Heterogeneous Order Arrival Rate Across Time

We also assume that orders may have different arrival rates throughout the shift. This section studies the impact of heterogenous order arrival rates across the 8-hour shift. For this purpose, we consider three scenarios: (1) homogeneous arrival rate: order arrival rate is the same throughout the shift, (2) heterogeneous arrival rate (type 1): the order arrival rate in the morning is 20% higher compared to the afternoon rate, (2) heterogeneous arrival rate (type 2): the afternoon rate is 20% higher than that during the morning. Table 6.9 reports an almost negligible performance difference (less than 0.5%) with respect to the distance traveled for the different strategies. However, with respect to completion time and tardiness, all systems, except the NPAs, perform better in the case of homogenous order arrival setup as opposed to the heterogenous setup. The trend is particularly pronounced in scenarios with higher order arrival rates (i.e., lambda equals 0.05). Finally, irrespective of the nature of the arrival rate, the proposed CHR-IPA(D), CHR-IPA(T), and CHR-RIP systems perform better than the benchmark models. However, comparing the proposed models,

Table 6.8: Impact of heterogeneous order arrival rates across zones on picking performance

Method	λ	Zone	Distance		Completion Time		Average Tardiness		Tardy Rate	
			μ	σ	μ	σ	μ	σ	μ	σ
H-IPA	0.01	homo	73.69	1.67	264.03	9.48	5.52	2.58	0.95	0.01
		hetero	71.71	4.07	373.87	21.89	32.17	13.33	0.88	0.03
	0.03	homo	47.19	1.26	354	13.36	17.37	3.97	0.9	0.01
		hetero	52.78	2.72	455.5	46.71	46.84	16.46	0.83	0.03
	0.05	homo	33.87	0.81	485.57	23.06	51.5	9.07	0.82	0.02
		hetero	35.63	2.07	771.93	109.77	174.66	63.91	0.66	0.06
CHR-IPA(D)	0.01	homo	5.35	0.22	105.33	5.53	2.06	1.41	0.98	0.01
		hetero	4.88	0.23	141.77	7.59	7.57	3.61	0.96	0.01
	0.03	homo	5.37	0.13	132.5	5.08	2.87	0.97	0.96	0.01
		hetero	5.04	0.16	158.77	6.42	6.79	1.48	0.95	0
	0.05	homo	5.29	0.09	185	9.91	7.77	1.72	0.93	0.01
		hetero	4.94	0.09	219.4	12.32	13.72	2.89	0.92	0.01
CHR-IPA(T)	0.01	homo	5.45	0.23	103.07	3.49	1.72	1.17	0.98	0.01
		hetero	7.71	0.48	202	13.95	15.62	6.69	0.92	0.03
	0.03	homo	5.49	0.13	129.43	3.3	2.46	0.65	0.96	0.01
		hetero	8.6	0.38	189.53	13.96	12.01	5.57	0.93	0.02
	0.05	homo	5.39	0.09	173.83	6.46	5.98	0.89	0.94	0.01
		hetero	8.35	0.24	264.33	27.22	24.23	9.06	0.89	0.02
CHR-RIPA	0.01	homo	5.37	0.2	96.97	4.04	2.07	1.21	0.98	0.01
		hetero	4.9	0.23	141.47	6.68	10.21	3.79	0.96	0.01
	0.03	homo	5.39	0.1	113.77	5.06	2.73	1.2	0.97	0.01
		hetero	5.05	0.17	131.03	7.4	6.37	2.45	0.96	0.01
	0.05	homo	5.22	0.07	187.33	13.29	10.05	3.41	0.93	0.01
		hetero	4.86	0.08	187.9	8.47	11.08	2.07	0.93	0.01
CHR-NPA(5)	0.01	homo	5.06	0.17	603.57	23.57	123.79	18.01	0.7	0.03
		hetero	6.51	0.39	958.83	72.19	363.75	75.6	0.53	0.05
	0.03	homo	5.11	0.12	362.33	5.64	30.93	3.18	0.83	0.01
		hetero	7.12	0.2	523.13	22.61	80.52	10.66	0.73	0.03
	0.05	homo	5.13	0.1	338.47	8.11	23.83	2.76	0.85	0.01
		hetero	7.05	0.12	433.77	11.04	44.89	5.62	0.79	0.02
CHR-NPA(C)	0.01	homo	2.62	0.07	2334.33	112.96	1474.89	117.11	0.16	0.03
		hetero	3.14	0.18	3444.57	324.35	2652.98	337.23	0.24	0.04
	0.03	homo	2.65	0.04	1163.07	22.85	399.72	24.42	0.38	0.02
		hetero	3.16	0.09	1760.03	96.6	950.68	102.84	0.24	0.02
	0.05	homo	2.67	0.03	917.67	14.28	228.08	12.35	0.51	0.02
		hetero	3.14	0.06	1310.53	47.2	517.4	38.91	0.32	0.03

we can observe that at higher overall order arrival rates, the homogenous scenarios perform substantial better in terms of completion time and tardiness. This can be attributed to the reasoning that impact of heterogenous arrival rates are possibly amplified when the overall rates are higher.

6.4 Implications and Discussion

In this section, we condense our findings and provide managerial guidelines to help warehouse professionals successfully implement CHR-OPS and achieve improved performance. Realizing that different warehouses face varied set of obstacles, we strive to provide customized recommendations based on the challenges faced.

To meet increased order volumes, despite labor shortages, some warehouses overload their workers with tasks, resulting in a serious injury rate, which is double the industry average (Gordon, 2021). Further, to ensure maximum productivity, specific warehouses even leverage technologies like "Pick by Voice," a system that provides item-picking instructions to the workers, to monitor workers' "time off task" (Gordon, 2021). Such practices have pressured pickers to work at dangerous speeds, resulting in increased injuries and skepticism about such technologies. To ameliorate the productivity expectations on the workers, gain their trust for automation systems, and avoid any legal consequences from worker overload, such warehouses can consider

Table 6.9: Impact of heterogeneous order arrival rates across time on picking performance

Method	λ	Distribution	Distance		Completion Time		Average Tardiness		Tard Rate	
			μ	σ	μ	σ	μ	σ	μ	σ
H-IPA	0.01	homo	73.69	1.67	264.03	9.48	5.52	2.58	0.95	0.01
		hetero	73.61	2.06	266.43	10.37	6.44	2.6	0.95	0.01
		hetero_op	73	2.09	267.6	8.93	6.24	2.39	0.95	0.01
	0.03	homo	47.19	1.26	354	13.36	17.37	3.97	0.9	0.01
		hetero	46.63	1.2	360.2	11.13	19.35	3.76	0.9	0.01
		hetero_op	46.47	1.06	361	13.04	19.89	3.57	0.89	0.01
	0.05	homo	33.87	0.81	485.57	23.06	51.5	9.07	0.82	0.02
		hetero	33.34	0.71	531.6	31.91	77.07	19.35	0.8	0.01
		hetero_op	33.19	0.93	543.7	36.84	84.09	25.95	0.79	0.02
CHR-IPA(D)	0.01	homo	5.35	0.22	105.33	5.53	2.06	1.41	0.98	0.01
		hetero	5.35	0.24	104.8	6.11	2.14	1.67	0.98	0.01
		hetero_op	5.36	0.23	106.57	5.87	2.25	1.61	0.98	0.01
	0.03	homo	5.37	0.13	132.5	5.08	2.87	0.97	0.96	0.01
		hetero	5.35	0.12	136	6.42	3.12	1.16	0.96	0.01
		hetero_op	5.37	0.14	137.13	5.68	3.41	1.19	0.96	0.01
	0.05	homo	5.29	0.09	185	9.91	7.77	1.72	0.93	0.01
		hetero	5.29	0.09	198.53	9.17	9.99	1.81	0.93	0.01
		hetero_op	5.27	0.09	198.2	11.51	10.4	2.41	0.92	0.01
CHR-IPA(T)	0.01	homo	5.45	0.23	103.07	3.49	1.72	1.17	0.98	0.01
		hetero	5.41	0.24	103.17	4.05	1.85	1.06	0.98	0.01
		hetero_op	5.42	0.23	103.9	3.94	1.82	1.43	0.98	0.01
	0.03	homo	5.49	0.13	129.43	3.3	2.46	0.65	0.96	0.01
		hetero	5.46	0.13	132.07	5.63	2.53	0.76	0.96	0.01
		hetero_op	5.49	0.15	132.13	3.69	2.69	0.79	0.96	0.01
	0.05	homo	5.39	0.09	173.83	6.46	5.98	0.89	0.94	0.01
		hetero	5.43	0.08	183.5	7.32	7.44	1.31	0.93	0.01
		hetero_op	5.38	0.07	183.6	6.94	7.69	1.2	0.93	0.01
CHR-RIPA	0.01	homo	5.37	0.2	96.97	4.04	2.07	1.21	0.98	0.01
		hetero	5.37	0.21	96.93	3.7	1.94	1.17	0.98	0.01
		hetero_op	5.39	0.21	98.8	5.27	2.64	1.58	0.98	0.01
	0.03	homo	5.39	0.1	113.77	5.06	2.73	1.2	0.97	0.01
		hetero	5.36	0.12	116.13	5.78	2.68	1.17	0.97	0.01
		hetero_op	5.36	0.13	119.47	5.02	3.16	1.23	0.97	0.01
	0.05	homo	5.22	0.07	187.33	13.29	10.05	3.41	0.93	0.01
		hetero	5.15	0.09	201.3	12.4	11.94	2.36	0.92	0.01
		hetero_op	5.16	0.09	212.63	19.38	15.57	5.1	0.92	0.01
CHR-NPA(5)	0.01	homo	5.06	0.17	603.57	23.57	123.79	18.01	0.7	0.03
		hetero	5.04	0.16	602.57	18.97	130.07	12.1	0.69	0.03
		hetero_op	5.06	0.17	607.13	25.76	135.19	22.55	0.68	0.03
	0.03	homo	5.11	0.12	362.33	5.64	30.93	3.18	0.83	0.01
		hetero	5.1	0.12	362.9	6.24	30.87	3.49	0.83	0.01
		hetero_op	5.08	0.13	363.43	6.03	30.98	2.84	0.83	0.01
	0.05	homo	5.13	0.1	338.47	8.11	23.83	2.76	0.85	0.01
		hetero	5.1	0.09	356.53	10.03	27.9	2.85	0.84	0.01
		hetero_op	5.08	0.09	358.4	12.62	28.56	3.66	0.83	0.01
CHR-NPA(C)	0.01	homo	2.62	0.07	2334.33	112.96	1474.89	117.11	0.16	0.03
		hetero	2.62	0.07	2316.27	146.83	1469.46	142.6	0.17	0.02
		hetero_op	2.61	0.07	2338.2	142.96	1486.14	145.18	0.16	0.02
	0.03	homo	2.65	0.04	1163.07	22.85	399.72	24.42	0.38	0.02
		hetero	2.65	0.04	1164.2	23.12	405.29	24.85	0.38	0.02
		hetero_op	2.65	0.04	1155.5	19.19	397.37	23.3	0.39	0.02
	0.05	homo	2.67	0.03	917.67	14.28	228.08	12.35	0.51	0.02
		hetero	2.66	0.03	918.37	11.57	229.66	10.24	0.51	0.02
		hetero_op	2.67	0.03	917.33	13.65	229.54	11.33	0.51	0.02

implementing CHR-OPS with the primary objective of reducing worker load.

Warehouse Type 1: Warehouses under pressure from limited workforce and increased worker injuries.

- Guideline 1: Implementing CHR-NPA(C), i.e., a collaborative human-robot order picking system following the no-intervention strategy and a policy that allows the robot to leave the depot only when the assigned pick list reaches the robot's full carrying capacity (C). CHR-NPA(C) is observed to reduce the travel distance of pickers by about half or multi-folds compared to the interventionist approaches to collaborative and manual picking, respectively.
- Guideline 2: Increase the number of zones. Increasing the number of zones is observed to reduce the picker travel distances significantly. Zoning ensures that the pickers are responsible for picking only within the designated regions, transferring most of the travel tasks to the robots. However, increasing the number of zones raises the minimum number of human pickers and robots required. Nevertheless, for warehouses facing many customer orders, more human pickers and robots may already be a necessity, and introducing increased zoning may not necessarily increase these requirements.
- Guideline 3: Increase the robot cart capacity. Increasing the AMR's cart capacity is also linked with reducing the picker travel distance. The decrease can be

attributed to more items being picked on a single trip, leading to fewer robot tours with many but closer picking positions, which in turn removes travel redundancies for the picker.

- Guideline 4: Implement warehouse slotting that ensures heterogenous order arrival rates across zones (i.e., store more frequently ordered items in the same zone). In warehouses with heterogenous zones (for example: in a two-zone system, one zone may have 20% higher incoming orders compared to the other), the new arriving orders are concentrated within the same zone, reducing the travel distance for human pickers.

On the contrary, few businesses solely depend on their warehouses' responsiveness. They employ enough human workers, and the main priority is to remain competitive by providing the quickest deliveries.

Warehouse Type 2: Warehouses that strive to achieve express deliveries.

- Guideline 1: Consider CHR-RIPA, i.e., a collaborative human-robot order picking system following the ruled based intervention strategy. CHR-RIPA is observed to provide the quickest completion times and lower tardiness. It is flexible to incorporate new orders into a picking tour, but only does so if the overall performance increases.

- Guideline 2: Implement warehouse slotting that ensures homogeneous order arrival rates across zones (i.e., store frequently ordered items with rarely picked ones in the same zone). In warehouses with homogeneous zones (for example: in a two-zone system, one zone may have 20% higher incoming orders compared to the order), the new arriving orders are distributed equally across the zones, this reduces picker waiting times and avoids congestions.

- Guideline 3: Increase the number of zones. Increasing the number of zones is observed to reduce both the picker and robot travel distances significantly. Thus, overall quicker completion times are achieved.

Warehouses may also hold highly customized products. Here large number of SKUs are stored, relative to the number of incoming orders. Further, since customer tend to pay more for customization, ensuring there are no delays becomes more critical.

Warehouse Type 3: Warehouses with low orders arrival rates.

- Guideline 1: Implement CHR-IPA(T), i.e., a collaborative human-robot order picking system following the intervention strategy and a policy that allocates a picker to a robot at the picking position based of the shortest time taken. For low order arrival rates, the CHR-IPA(T) is observed to provide least average tardiness among all the systems.

- Guideline 2: Investing in economic robots with limited capacity is sufficient. Increasing the cart size does not significantly improve performance in scenarios with low order arrival rates. The trend can be attributed to the fact that robots frequently travel back to the depot and unload in such scenarios.

6.5 Conclusions

In this chapter, we study a collaborative human-robot order picking system (CHR-OPS), where human pickers retrieve ordered items from storage aisles and transport robots carry them from the pick location to the depot. Unlike extant research, this study proposes interventionist and rule-based interventionist approaches to CHR-OPS to help the system process dynamic customer orders. Further, the proposed algorithms are benchmarked against traditional manual picking with interventions and CHR-OPS with no interventions. The performances are compared using holistic measures like distance traveled, completion time, and tardiness. Our results indicate that the interventionist and rule-based interventionist approaches perform the best in terms of completion time and tardiness, which are most critical to today's competitive warehouses. Notably, the interventionist approaches to CHR-OPS perform 5 to 22 times better than CHR-OPS with no interventions in terms of completion times and tardiness. Further, the study conducts numerous experiments to analyze the impact of aspects like zoning, robot capacity, and the nature of order arrivals. Based on

the results obtained, managerial insights personalized to the objective of a warehouse are provided. For instance, warehouses seeking to gain a competitive advantage by offering express deliveries are advised to consider implementing a rule-based interventionist algorithm for their CHR-OPS while ensuring slotting that enables homogenous order arrival rates across the warehouse zones.

For future work, the proposed interventionist algorithms for collaborative order picking can be extended in numerous ways. First, we can modify the algorithm to implement it in a multi-block warehouse layout found in large scale warehouses. Second, the proposed CHR-OPSs can be studied under different warehouse storage strategies, like random storage and dedicated storage, for a comparative analysis. Third, CHR-OPS can be extended to handle re-stocking operations as well, for example in warehouses with dedicated fast-pick areas. Finally, future research can also try implementing the CHR-OPS for cross docking, where items need to be retrieved from inbound transportation vehicles (instead of picking aisles) and loaded onto assigned outbound vehicles (instead of a single warehouse depot).

Chapter 7

Conclusions and Future Work

Order picking is a critical function in warehouses, accounting for a significant portion of operating costs. Inefficient planning of picking operations can lead to suboptimal asset utilization and delayed deliveries, negatively impacting customer satisfaction and competitiveness. To address these challenges, warehouses are exploring collaborative human-robot order-picking systems (CHR-OPS) using autonomous mobile robots (AMRs) or collaborative robots (cobots) to improve efficiency and reduce labor intensity. This research aims to optimize key decisions in two warehouse picking strategies: static picking and dynamic picking.

For static picking, where the items to be retrieved are known in advance, the study focuses on optimizing order batching, batch assignment and sequencing, and picker-

robot routing decisions. A mixed integer linear programming model is developed to minimize total order tardiness, and efficient solution algorithms are proposed. Results show the impact of factors like human-robot team composition, AMR speed, capacity, and warehouse layout on picking efficiency.

For dynamic picking, where orders arrive over time, two interventionist policies are proposed: CHR-IPA and CHR-RIPA. These policies allow for dynamic rescheduling of missions and evaluate new orders based on their impact on average moving distance per item. The evaluation demonstrates improved system performance compared to benchmark approaches, with CHR-IPA outperforming CHR-RIPA in terms of average tardiness and order completion time.

The research provides managerial insights for implementing collaborative order picking systems and highlights the adaptability of the proposed models to various warehouse settings. The study also identifies future research directions to further enhance the efficiency and effectiveness of CHR-OPS.

This dissertation also identified directions for future work. First, order picking with re-stocking process. Investigating the integration of order picking with the re-stocking process can be an interesting avenue for research. This involves examining how the availability of items for picking is influenced by the re-stocking activities in the warehouse. By considering the dynamic nature of inventory replenishment, it would be valuable to develop strategies that optimize both the order picking process

and the efficient replenishment of items to ensure continuous availability. Second, random storage. Another interesting direction for future research is to explore the implications of random storage strategies in the warehouse. Random storage allows items to be stored in any available location, rather than having a predefined location for each item. This approach can introduce additional flexibility and adaptability in the picking process. Investigating the impact of random storage on order picking efficiency, resource utilization, and overall warehouse performance would be beneficial in understanding its potential advantages and challenges. Third, conduct real-world experiments to validate the findings. While simulation studies provide valuable insights, conducting real-world experiments is crucial to validate the effectiveness and practicality of proposed order picking strategies. Real-world experiments allow for the evaluation of the proposed approaches in authentic warehouse settings, considering various operational constraints and complexities. Gathering empirical data and analyzing the performance metrics in real-time scenarios would provide valuable evidence to support the efficacy and feasibility of the proposed strategies.

References

Ardjmand, E., Bajgiran, O. S., & Youssef, E. (2019). Using list-based simulated annealing and genetic algorithm for order batching and picker routing in put wall based picking systems. *Applied Soft Computing Journal*, 75, 106–119. https://doi.org/10.1016/J.ASOC.2018.11.019

Ardjmand, E., Shakeri, H., Singh, M., & Sanei Bajgiran, O. (2018). Minimizing order picking makespan with multiple pickers in a wave picking warehouse. *International Journal of Production Economics*, 206, 169–183. https://doi.org/https://doi.org/10.1016/j.ijpe.2018.10.001

Azadeh, K., Roy, D., & de Koster, M. R. (2020). Dynamic human-robot collaborative picking strategies. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3585396

Bahçeci, U. & Öncan, T. (2021). An evaluation of several combinations of routing and storage location assignment policies for the order batching problem. *International Journal of Production Research*, 1–20.

Caputo, A. C. & Pelagagge, P. M. (2006). Management criteria of automated order picking systems in high-rotation high-volume distribution centers. *Industrial Management & Data Systems*, 106, 1359–1383. https://doi.org/10.1108/02635570610712627

Chen, T.-L., Cheng, C.-Y., Chen, Y.-Y., & Chan, L.-K. (2015). An efficient hybrid algorithm for integrated order batching, sequencing and routing problem. *International Journal of Production Economics*, 159, 158–167. https://doi.org/https://doi.org/10.1016/j.ijpe.2014.09.029

Cheng, C.-Y., Chen, Y.-Y., Chen, T.-L., & Yoo, J. J.-W. (2015). Using a hybrid approach based on the particle swarm optimization and ant colony optimization to solve a joint order batching and picker routing problem. *International Journal of Production Economics*, 170, 805–814.

de Koster, R., Le-Duc, T., & Roodbergen, K. J. (2007). Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182, 481–501. https://doi.org/10.1016/j.ejor.2006.07.009

Drury, J. (1988). Towards more efficient order picking. *IMM monograph*, 1(1), 1–69.

Fager, P., Sgarbossa, F., & Calzavara, M. (2021). Cost modelling of onboard cobot-supported item sorting in a picking system. *International Journal of Production Research*, 59, 3269–3284. https://doi.org/10.1080/00207543.2020.1854484

Ferreira, C., Figueira, G., & Amorim, P. (2021). Scheduling human-robot teams in

collaborative working cells. *International Journal of Production Economics*, 235. https://doi.org/10.1016/j.ijpe.2021.108094

Gademann, N. & van de Velde, S. (2005). Order batching to minimize total travel time in a parallel-aisle warehouse. *IIE Transactions (Institute of Industrial Engineers)*, 37, 63–75. https://doi.org/10.1080/07408170590516917

Goetschalckx, M. & Ashayer, J. (1989). Classification and design of order picking. *Logistics World*, 2, 99–106. https://doi.org/10.1108/EB007469

Henn, S. (2015). Order batching and sequencing for the minimization of the total tardiness in picker-to-part warehouses. *Flexible Services and Manufacturing Journal*, 27, 86–114. https://doi.org/10.1007/s10696-012-9164-1

Ho, Y.-C., Su, T.-S., & Shi, Z.-B. (2007). Order-batching methods for an order-picking warehouse with two cross aisles. *Computers & Industrial Engineering*. https://doi.org/10.1016/j.cie.2007.12.018

Jaghbeer, Y., Hanson, R., & Johansson, M. I. (2020). Automated order picking systems and the links between design and performance: a systematic literature review. *International Journal of Production Research*, 4489–4505. https://doi.org/10.1080/00207543.2020.1788734

Lee, H. Y. & Murray, C. C. (2019). Robotics in order picking: evaluating warehouse layouts for pick, place, and transport vehicle routing systems. *International Journal*

of Production Research, 57, 5821–5841. https://doi.org/10.1080/00207543.2018.1552031

Liu, H. & Wang, L. (2020). Remote human–robot collaboration: A cyber–physical system application for hazard manufacturing environment. *Journal of Manufacturing Systems*, 54, 24–34. https://doi.org/10.1016/J.JMSY.2019.11.001

Löffler, M., Boysen, N., & Schneider, M. (2021). Picker routing in agv-assisted order picking systems. *INFORMS Journal on Computing*. https://doi.org/10.1287/ijoc.2021.1060

Napolitano, M. (2012). 2012 warehouse/dc operations survey: mixed signals. *Logistics management (Highlands Ranch, Colo.: 2002)*, 51(11).

Pan, J. C. H., Wu, M. H., & Chang, W. L. (2014). A travel time estimation model for a high-level picker-to-part system with class-based storage policies. *European Journal of Operational Research*, 237, 1054–1066. https://doi.org/10.1016/J.EJOR.2014.02.037

Parthasarathy, S. & Rajendran, C. (1997). A simulated annealing heuristic for scheduling to minimize mean weighted tardiness in a flowshop with sequence-dependent setup times of jobs-a case study. *Production Planning & Control*, 8(5), 475–483.

Rahman, S. M. & Wang, Y. (2015). Dynamic affection-based motion control of a humanoid robot to collaborate with human in flexible assembly in manufacturing.

ASME 2015 Dynamic Systems and Control Conference, DSCC 2015, 3. https://doi.org/10.1115/DSCC2015-9841

Ratliff, H. D. & Rosenthal, A. S. (1983). Order-picking in a rectangular warehouse: A solvable case of the traveling salesman problem. *Operations Research*, 31, 507–521. https://doi.org/10.1287/OPRE.31.3.507

Scholz, A., Schubert, D., & Wäscher, G. (2017). Order picking with multiple pickers and due dates – simultaneous solution of order batching, batch assignment and sequencing, and picker routing problems. *European Journal of Operational Research*, 263, 461–478. https://doi.org/10.1016/j.ejor.2017.04.038

Scholz, A. & Wäscher, G. (2017). Order batching and picker routing in manual order picking systems: the benefits of integrated routing. *Central European Journal of Operations Research*, 25(2), 491–520.

Tompkins, J. A., White, J. A., Bozer, Y. A., & Tanchoco, J. M. A. (2010). *Facilities planning*. John Wiley & Sons.

Çağla Cergibozan & Tasan, A. S. (2019). Order batching operations: an overview of classification, solution techniques, and future research. *Journal of Intelligent Manufacturing*, 30, 335–349. https://doi.org/10.1007/S10845-016-1248-4

Žulj, I., Salewski, H., Goeke, D., & Schneider, M. (2021). Order batching and batch sequencing in an amr-assisted picker-to-parts system. *European Journal of Operational Research*. https://doi.org/10.1016/j.ejor.2021.05.033

Vita

Shitao Yu was born in China, Henan province. He attended Henan Experimental Mid & High School in Zhengzhou, Henan. He completed his Bachelor of Engineering degree in Industrial Engineering, at Huazhong University of Science and Technology at Wuhan in 2016, while during 2015 to 2016 he studied in University of Missouri as an exchange student. He then stayed in University of Missouri, where he earned his Master's in 2019 under the advisement of Dr. Noble James and Ph.D. in 2023 under the advisement of Dr. Sharan Srinivas.