Int. J. Production Economics 254 (2022) 108634

Contents lists available at ScienceDirect

International Journal of Production Economics

journal homepage: www.elsevier.com/locate/ijpe

Collaborative order picking with multiple pickers and robots: Integrated approach for order batching, sequencing and picker-robot routing

Sharan Srinivas [a,b,c,*], Shitao Yu [a]

[a] Department of Industrial and Manufacturing Systems Engineering, College of Engineering, University of Missouri, Columbia, MO 65211, United States of America
[b] Department of Marketing, Trulaske College of Business, University of Missouri, Columbia, MO 65211, United States of America
[c] Institute for Data Science and Informatics, University of Missouri, Columbia, MO 65211, United States of America

ARTICLE INFO

Keywords:
Collaborative order picking
Autonomous mobile robots
Order batching
Picker-robot routing
Optimization
Simulated annealing

ABSTRACT

With the growing demand for speed and flexibility in order fulfillment, it is crucial to employ an efficient picking system that allows faster delivery of items from warehouse storage to a depot. The use of autonomous mobile robots (AMRs) for intra-logistics can improve picking productivity, while alleviating the strain on human workers. In this research, a collaborative human–robot order-picking system (CHR-OPS) is considered, where humans perform item retrieval tasks and AMRs handle item transportation to the depot. The delivery performance of a CHR-OPS for a given set of orders and their due dates depend on three subproblems: (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 system employing multiple pickers and AMRs. An optimization model is developed to jointly optimize the three problems with the objective of minimizing the total tardiness of all orders. A simulated annealing algorithm with adaptive neighborhood search and optimization-based restart strategy is proposed for handling large instances. 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.

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. Automation, enabled by Industry 4.0 technologies, provides the capability to reduce the dependence on labor resources and improve overall warehouse operations (Caputo and 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

* Correspondence to: E3437 Lafferre Hall, University of Missouri, Columbia, MO 65211, USA.
E-mail addresses: SrinivasSh@missouri.edu (S. Srinivas), sy5wf@mail.missouri.edu (S. Yu).

https://doi.org/10.1016/j.ijpe.2022.108634
Received 25 January 2022; Received in revised form 29 June 2022; Accepted 7 September 2022
Available online 13 September 2022

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 and Murray, 2019). While specialized picker robots may be used to complement the AMRs, such a fully automated OPS becomes a capital-intensive operation (Goetschalckx and Ashayer, 1989; Drury, 1988; 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. Fig. 1 shows the two types of robots used in the follow-pick and swarm CHR-OPS.

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 minimize the total tardiness time 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 robot-picker 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.
- 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.
- 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, 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.
- 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.

The remainder of this paper is organized as follows. A review of relevant literature pertaining to both traditional and collaborative order picking is presented in Section 2. Section 3 describes the problem in detail and presents the MILP formulation for the CHR-OBASRP. The solution approach for large instances based on the proposed RSA-ANS is described in Section 4. The numerical analysis of the proposed models is presented in Section 5. Finally, the conclusions, insights on CHR-OPS, and future research directions are provided in Section 6.

2. Related work and contributions

Given the importance of order picking, 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 and Wang, 2015), hazard manufacturing environment (Liu and 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 the relevant works in CHR-OPS. Subsequently, we also discuss 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.

(a) (b)

Fig. 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].

2.1. 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 and 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 and 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 algorithm 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.

2.2. 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 (Gademann and van de Velde, 2005; Bahçeci and Öncan, 2021). 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 Cergibozan and 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., Henn, 2015; Cheng et al., 2015; Chen et al., 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.

Table 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
This paper	**tardiness**	✓	✓	✓	✓	**M**	**M**	**M**	**Y**

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

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.

2.3. Research gaps and contributions

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 and 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 SA 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 contributes to the literature on OPS by developing optimization models and efficient solution approaches for the CHR-OBASRP. Table 1 summarizes the relevant works on OPS and highlights the contributions of this research.

3. Collaborative human–robot order picking system

3.1. Problem description

In this research, 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 Fig. 2. 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 2 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., $1, 2, \ldots, |\mathcal{N}|$), 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

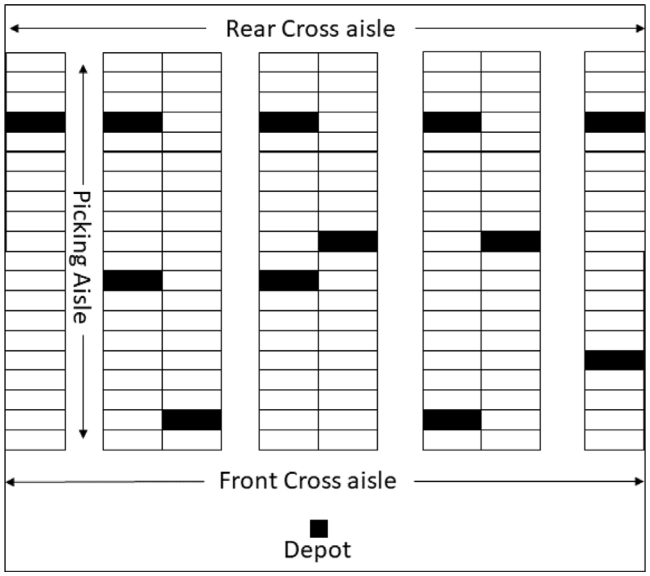

Fig. 2. Illustration of Block Warehouse Layout.

Table 2
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	100 001	1	2	A01-R01-S02-P05
	230 653	2	1	A04-R02-S01-P08
	242 411	3	2	A04-R04-S01-P03
2	629 360	4	1	A14-R02-S03-P10
	100 001	5	1	A01-R01-S02-P05
⋮	⋮	⋮	⋮	⋮
$\mathcal{J}$	365 312	$\mathcal{N}-1$	1	A06-R05-S02-P08
	242 411	$\mathcal{N}$	2	A04-R04-S01-P03

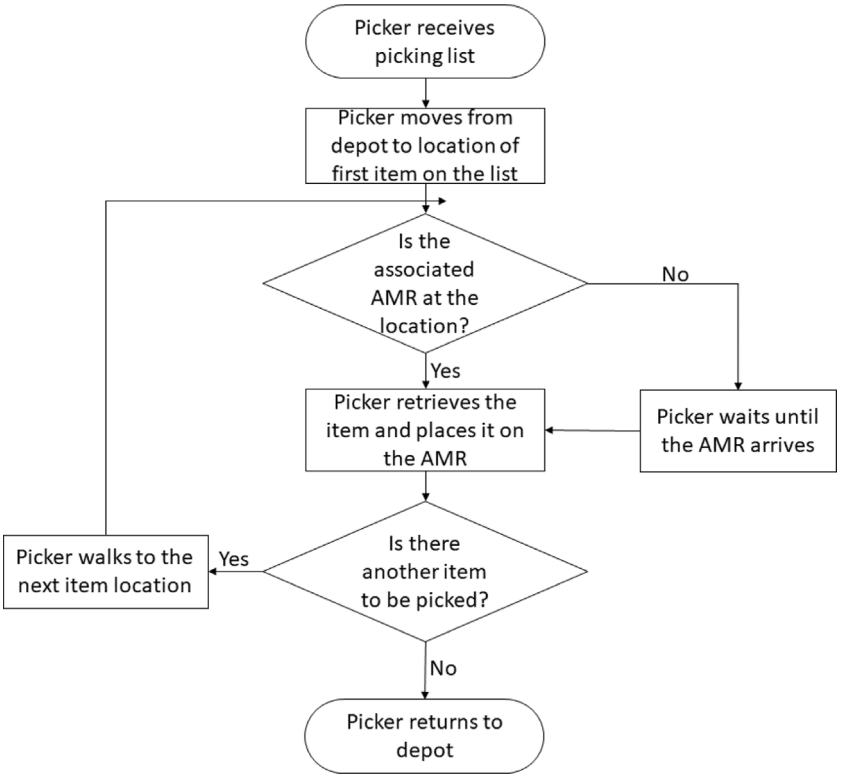

Fig. 3. Pick-and-place process from human picker's perspective in a CHR-OPS.

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 2). 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.

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 $\langle \tau_{0i}^R, \tau_{i0}^R \rangle$ and $\langle \tau_{0i}^K, \tau_{i0}^K \rangle$ 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 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. Figs. 3 and 4 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 and 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.

3.2. Mathematical model for collaborative order picking

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), $\mathcal{R} = \{r_1, r_2, \ldots, r_{\max}\}$
$k \in \mathcal{K}$	Set of human pickers, $\mathcal{K} = \{k_1, k_2, \ldots, k_{\max}\}$
$t \in \mathcal{T}$	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 tripbut 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

The model is mathematically formulated as follows.

$$\min \quad Z = \sum_{j \in \mathcal{J}} T_j \tag{1}$$

subject to

$$\sum_{k \in \mathcal{K}} A^K_{i,k} = 1 \qquad \forall i \in \mathcal{N} \tag{2}$$

$$\sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}^R} A^R_{i,r,t} = 1 \qquad \forall i \in \mathcal{N} \tag{3}$$

$$X^R_{r,t} \le \sum_i A^R_{i,r,t} r \in \mathcal{R}, \qquad \forall t \in \mathcal{T} \tag{4}$$

$$\sum_i \beta_i \cdot A^R_{i,r,t} \le \kappa_r \cdot X^R_{r,t} \qquad r \in \mathcal{R}, \forall t \in \mathcal{T} \tag{5}$$

$$X^R_{r,t} \ge X^R_{r,t+1} \qquad r \in \mathcal{R}, t \in \mathcal{T}, t < |\mathcal{T}| \tag{6}$$

$$A^K_{i,k} - A^K_{i',k} \le 1 - (U^K_{i,i'} + U^K_{i',i}) \qquad i, i' \in \mathcal{N}, i \ne i', k \in \mathcal{K} \tag{7}$$

$$A^K_{i,k} + A^K_{i',k} \le 1 + (U^K_{i,i'} + U^K_{i',i}) \qquad i, i' \in \mathcal{N}, i \le i', k \in \mathcal{K} \tag{8}$$

$$A^R_{i,r,t} - A^R_{i',r,t} \le 1 - (U^R_{i,i'} + U^R_{i',i}) \qquad i, i' \in \mathcal{N}, i \ne i', r \in \mathcal{R}, t \in \mathcal{T} \tag{9}$$

$$A^R_{i,r,t} + A^R_{i',r,t} \le 1 + (U^R_{i,i'} + U^R_{i',i}) \qquad i, i' \in \mathcal{N}, i \le i', r \in \mathcal{R}, t \in \mathcal{T} \tag{10}$$

$$B^K_{i,k} \ge S^K_k + \tau^K_{0i} - M \cdot (1 - A^K_{i,k}) \qquad i \in \mathcal{N}, k \in \mathcal{K} \tag{11}$$

$$\sum_{k \in \mathcal{K}} B^K_{i,k} \ge \sum_{k \in \mathcal{K}} L^K_{i',k} + \tau^K_{i',i} \cdot U^K_{i',i} - M \cdot (1 - U^K_{i',i}) \qquad i, i' \in \mathcal{N}, k \in \mathcal{K}, i \ne i' \tag{12}$$

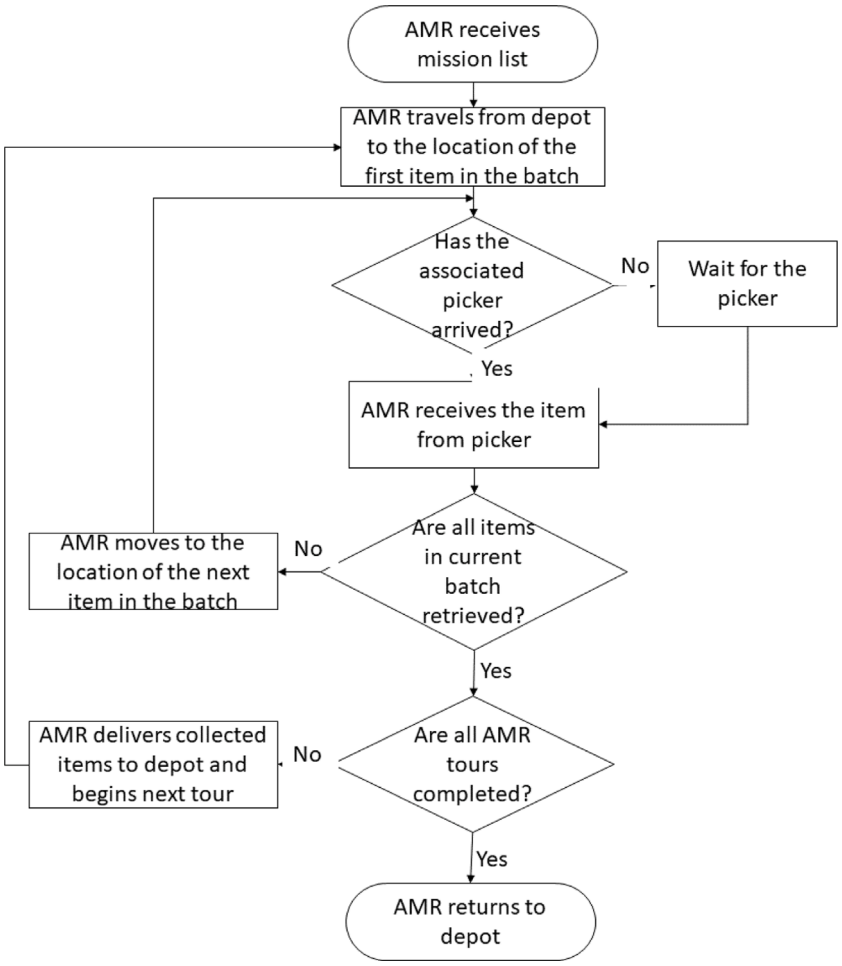

Fig. 4. Item collection process from AMR's perspective in a CHR-OPS.

$$F^K_{i,k} = B^K_{i,k} + \eta^U_{i,k} \cdot A^K_{i,k} \qquad i \in \mathcal{N}, k \in \mathcal{K} \qquad (13)$$

$$L^K_{i,k} \geq F^K_{i,k} \qquad i \in \mathcal{N}, k \in \mathcal{K} \qquad (14)$$

$$L^K_{i,k} \geq \sum_{t \in \mathcal{T}} F^R_{i,r,t} - M \cdot (2 - A^K_{i,k} - \sum_{t \in \mathcal{T}} A^R_{i,r,t}) \qquad i \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R} \qquad (15)$$

$$B^K_{i,k} \leq M \cdot A^K_{i,k} \qquad i \in \mathcal{N}, k \in \mathcal{K} \qquad (16)$$

$$F^K_{i,k} \leq M \cdot A^K_{i,k} \qquad i \in \mathcal{N}, k \in \mathcal{K} \qquad (17)$$

$$L^K_{i,k} \leq M \cdot A^K_{i,k} \qquad i \in \mathcal{N}, k \in \mathcal{K} \qquad (18)$$

$$B^R_{i,r,t} \geq S^R_{r,t} + \tau^R_{0i} - M \cdot (1 - A^R_{i,r,t}) \qquad i \in \mathcal{N}, r \in \mathcal{R}, t \in \mathcal{T} \qquad (19)$$

$$\sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}^R} B^R_{i,r,t} \geq \sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}^R} F^R_{i',r,t} + \tau^R_{i',i} \cdot U^R_{i',i} - M \cdot (1 - U^R_{i',i}) \qquad i, i' \in \mathcal{N}, r \in \mathcal{R}, i \neq i' \qquad (20)$$

$$\sum_{t \in \mathcal{T}} B^R_{i,r,t} \geq \sum_{k \in \mathcal{K}} F^K_{i,k} - M \cdot (2 - \sum_{k \in \mathcal{K}} A^K_{i,k} - \sum_{t \in \mathcal{T}} A^R_{i,r,t}) \qquad i \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R} \qquad (21)$$

$$F^R_{i,r,t} = B^R_{i,r,t} + \eta^L_{i,r} \cdot A^R_{i,r,t} \qquad \forall i \in \mathcal{N}, r \in \mathcal{R}, t \in \mathcal{T} \qquad (22)$$

$$B^R_{i,r,t} \leq M \cdot A^R_{i,r,t} \qquad i \in \mathcal{N}, r \in \mathcal{R}, t \in \mathcal{T} \qquad (23)$$

$$F^R_{i,r,t} \leq M \cdot A^R_{i,r,t} \qquad i \in \mathcal{N}, r \in \mathcal{R}, t \in \mathcal{T} \qquad (24)$$

$$C^R_{r,t} \geq F^R_{i,r,t} + \tau^R_{i0} - M \cdot (1 - A^R_{i,r,t}) \qquad i \in \mathcal{N}, r \in \mathcal{R}, t \in \mathcal{T} \qquad (25)$$

$$S^R_{r,t} \geq C^R_{r,t-1} - M \cdot (1 - X^R_{r,t}) \qquad t \in \mathcal{T}, r \in \mathcal{R} \qquad (26)$$

$$C_j \geq C^R_{r,t} - M \cdot (1 - A^R_{i,r,t}) \qquad \forall i \in \mathcal{O}_j, t \in \mathcal{T}, r \in \mathcal{R}, j \in \mathcal{J} \qquad (27)$$

$$T_j \geq C_j - d_j \qquad \forall j \in \mathcal{J} \qquad (28)$$

$$S^R_{r,t} \leq M \cdot X^R_{r,t} \qquad r \in \mathcal{R}, t \in \mathcal{T} \qquad (29)$$

$$C^R_{r,t} \leq M \cdot X^R_{r,t} \qquad r \in \mathcal{R}, t \in \mathcal{T} \qquad (30)$$

$$A^K_{i,k}, A^R_{i,r,t}, X^R_{r,t}, U^K_{i',i}, U^R_{i',i} \in \{0,1\} \qquad i, i' \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R}, t \in \mathcal{T} \qquad (31)$$

$$B^K_{i,k}, F^K_{i,k}, L^K_{i,k}, F^R_{i,r,t}, B^R_{i,r,t}, C^K_k, C^R_{r,t}, S^K_k, S^R_{r,t}, C_j, T_j \geq 0 \qquad i \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R}, t \in \mathcal{T}, j \in \mathcal{J} \qquad (32)$$

The MILP model for CHR-OBASRP is given by Eqs. (1)–(32). The objective function (Eq. (1)) seeks to minimize the total tardiness. Constraint (2) ensures that the pick location of every item to be retrieved is assigned to be visited by exactly one operator, while constraint (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 (4)–(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 (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 (5) will force that AMR to make a non-empty tour t (i.e., $X^R_{r,t} = 1$). In addition, constraint (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 (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 (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 (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 (7)–(8). Therefore, if items i and i' are assigned to the same operator k, then Constraints (7)–(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 (9)–(10).

The retrieval begin time of an item from storage is controlled by constraints (11) and (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 (13). The earliest time at which an operator can leave a pick location is given by constraints (14) and (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 (16)–(18).

Likewise, the start time of placing item i on an AMR is governed by constraints (19)–(21). Specifically, the AMR cannot collect the item before it arrives to the pick location of that item, as given by constraints (19) and (20). In addition, the AMR can collect the item only after the human worker completes the pick operation (constraint (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 (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 (23)–(24).

The AMR tour completion time (equivalently batch completion time) is determined by constraint (25). On the other hand, constraint (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 (26) for all AMRs. The delivery completion time of all the items in an order is given by constraint (27), and the corresponding tardiness is determined by constraint (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 (30). Finally, the non-negativity and binary restrictions on the decision variables are specified by constraints (31)–(32), respectively.

4. Solution approach

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 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 v^0 at temperature $\theta^{\max}$, and the associated objective function value is $f(v^0)$. 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 Section 4.2. To initiate the iterative search process, the current solution (v), objective ($f(v)$) and temperature (θ) values are set as v^0, $f(v^0)$, $\theta^{\max}$, respectively. In addition, the best found solution is also initialized ($v^* \leftarrow v^0$, $f(v^*) \leftarrow f(v^0)$). At a given iteration I and temperature θ, a candidate solution v' with objective $f(v')$ is generated by exploring the neighborhood solutions of v. For a minimization objective, if $f(v') \leq f(v)$, then the neighboring solution is set as the incumbent and best found solution (i.e., $v \leftarrow v'$ and $v^* \leftarrow v'$). 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 $g \sim (0,1) < e^{(f(v)-f(v'))/\theta}$. 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) $f(v)$ remains unchanged for the $\rho^{\max}$ iterations of the search procedure (iii) $f(v)$ 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 1. 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 4.3), 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 4.4 and 4.5, 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 and 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 4.6. Finally, the RSA-ANS returns the best-found solution and the corresponding tardiness value (line 40).

Algorithm 1 Restarted Simulated Annealing with Adaptive Neighborhood Search

1: **Inputs:** problem parameters, θ^0, θ^{min}, α, I^{max}, ρ^{max}, ρ^{RS}, π, N_l where $l = \{1, 2, ..., l^{max}\}$
2: generate initial solution (i.e., mission and pick lists), ν^0
3: initialize $\nu \leftarrow \nu^0$, $\nu^* \leftarrow \nu^0$, $\theta \leftarrow \theta^0$, $\rho \leftarrow 0$
4: compute $f(\nu)$ and $f(\nu^*)$
5: **while** $\theta > \theta^{min}$ **and** $\rho < \rho^{max}$ **and** $f(\nu^*) \neq 0$ **do**
6: $I \leftarrow 1$
7: $p_l = \frac{1}{l^{max}}, \forall l = \{1, 2, ..., l^{max}\}$
8: **while** $I \leq I^{max}$ **and** $f(\nu^*) \neq 0$ **do**
9: choose π operators from $\{N_1, N_2, ..., N_{l^{max}}\}$ based on selection probability p_l
10: generate neighborhood solutions, $\nu_n, n = 1, 2, ..., \pi$, using selected operators
11: compute tardiness of all neighborhood solutions, $f(\nu_n), n = 1, 2, ..., \pi$
12: $f(\nu') \leftarrow \min\{f(\nu_1), f(\nu_2), ..., f(\nu_\pi)\}$ and $\nu' \leftarrow \nu'_{\text{argmin}\{f(\nu_1), f(\nu_2), ..., f(\nu_\pi)\}}$
13: **if** $f(\nu') < f(\nu)$ **then**
14: $\nu \leftarrow \nu'$ and $f(\nu) \leftarrow f(\nu')$
15: **else**
16: $\Delta = \frac{f(\nu') - f(\nu)}{f(\nu)}$
17: generate random number $g \sim U(0, 1)$
18: **if** $g \leq e^{-\Delta/\theta}$ **then**
19: update $\nu \leftarrow \nu'$ and $f(\nu) \leftarrow f(\nu')$
20: **end if**
21: **if** $f(\nu') < f(\nu^*)$ **then**
22: $\nu^* \leftarrow \nu'$ and $f(\nu^*) \leftarrow f(\nu')$
23: $\rho \leftarrow 0$
24: **else**
25: $\rho \leftarrow \rho + 1$
26: **end if**
27: adjust p_l of neighborhood operator N_l, where $l = \{1, 2, ..., l^{max}\}$ (see Section 4.4)
28: **end if**
29: $I \leftarrow I + 1$
30: **end while**
31: $\theta \leftarrow \alpha \times \theta$
32: **if** $\rho \geq \rho^{RS}$ **then**
33: Use *fix-and-optimize* heuristic to find a new restart point ν'' (see Section 4.6)
34: **if** $f(\nu'') \neq f(\nu)$ **then**
35: $\nu \leftarrow \nu''$ and $f(\nu) \leftarrow f(\nu'')$
36: $\rho \leftarrow 0$
37: **end if**
38: **end if**
39: **end while**
40: **return** $f(\nu^*)$ and ν^*

4.2. Solution representation of CHR-OBASRP

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 $k \in \mathcal{K}$. Fig. 5 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 $i_{[end]}$. 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.

Fig. 6 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 Fig. 6, 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.

4.3. 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.

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.

4.4. 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.

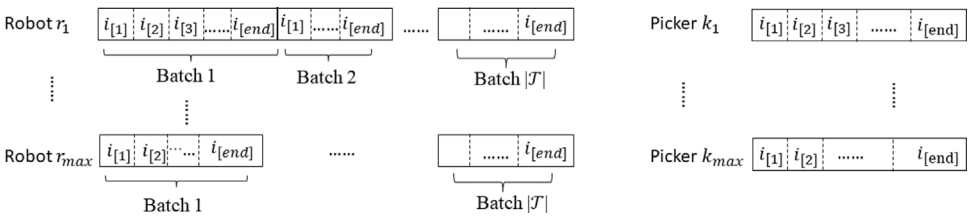

Fig. 5. General Illustration Solution of CHR-OBASRP.

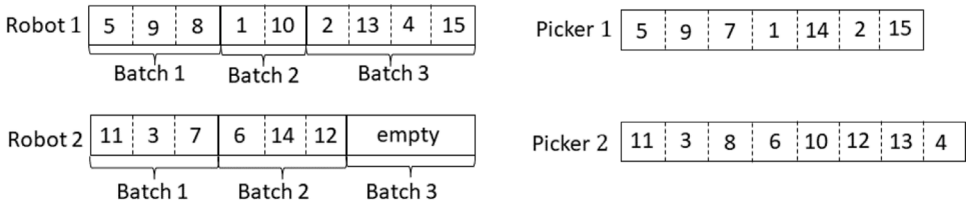

Fig. 6. Numerical illustration of mission and pick lists.

Algorithm 2 Rule-based heuristic for CHR-OBASRP

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
Output: mission lists ($\mathcal{M}_{rt}$, $\forall r \in \mathcal{R}, t \in \mathcal{T}$) and pick lists ($\mathcal{P}_k$, $\forall k \in \mathcal{K}$)
Set total items assigned for robot r in trip t to zero (i.e., $\bar{I}_{r,t} = 0 \ \ \forall r \in \mathcal{R}, \forall t \in \mathcal{T}$)
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, $\forall r \in \mathcal{R}$)
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
while $\mathcal{L}^{\mathcal{R}} \neq \emptyset$ **do**
  Select the first item i_1 from $\mathcal{L}$
  for each operator $k \in \mathcal{K}$ **do**
    Estimate $B^K_{i_1,k}$ and $F^K_{i_1,k}$
  end for
  Assign item i_1 to operator $k^* = \underset{k\in\mathcal{K}}{\operatorname{argmin}}\{B^K_{i_1,k}\}$
  Append item i_1 to the end of $\mathcal{P}_{k^*}$
  for each robot $r \in \mathcal{R}$ **do**
    Select current trip t^*_r
    if $\bar{I}_{r,t} < \kappa_r$ **then**
      Compute collection begin time for robot r in trip t, $B^R_{i_1,r,t^*_r}$
    else
      if $t^*_r < |\mathcal{T}|$ **then**
        Set trip indicator $t^*_r = t^*_r + 1$
        Compute collection begin time for robot r in trip t^*_r, $B^R_{i_1,r,t^*_r}$
      else
        Set $B^R_{i_1,r,t^*_r} = M$, where M is a large positive number
      end if
    end if
  end for
  Assign item i_1 to robot $r^* = \underset{r\in\mathcal{R}}{\operatorname{argmin}}\{B^R_{i_1,r,t^*_r}\}$
  Append item i_1 to the end of $\mathcal{M}_{r^*,t^*_r}$
  Remove item i_1 from the list $\mathcal{L}$.
end while
return $\mathcal{M}_{r,t}$ and $\mathcal{P}_k$

- 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.

Figs. 7 and 8 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.

4.5. Adaptive selection of neighborhood operators

Given a set of $l^{\max}$ neighborhood structures ($N_1, N_2, \ldots, N_{l^{\max}}$), 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

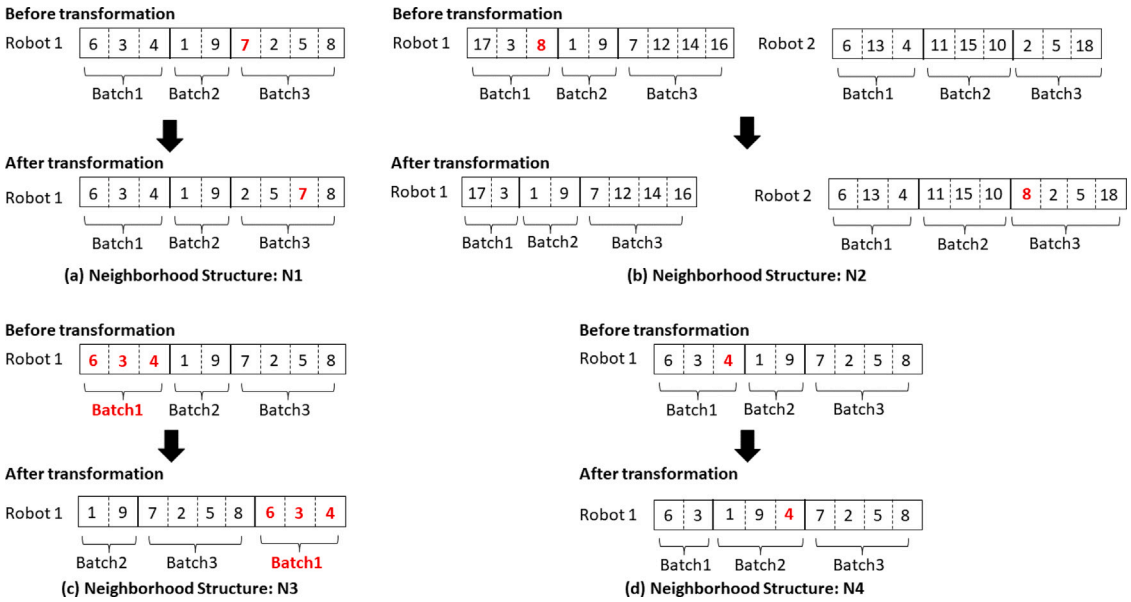

Fig. 7. Illustration of neighborhood structures that uses move operator ($N_1 - N_4$).

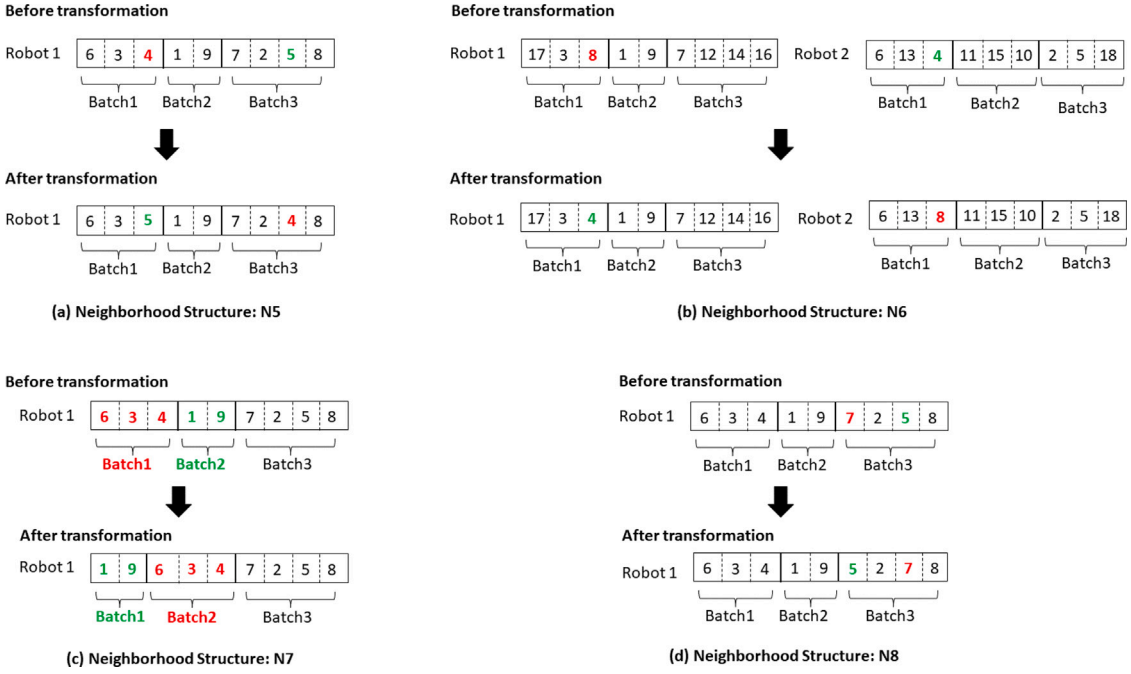

Fig. 8. Illustration of neighborhood structures that uses swap operator ($N_5 - N_8$).

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 Eq. (33), 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.

$$w_l = \xi + (1 - l^{\max} \times \xi) \times \frac{\phi_l}{\sum_l \phi_l} \qquad \forall l = 1, 2, \ldots, l^{\max} \tag{33}$$

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 Eq. (34). Note that the operators are selected without replacement, and therefore, the same operator cannot be chosen twice.

$$p_l = \frac{w_l}{\sum_l w_l} \qquad \forall l = 1, 2, \ldots, l^{\max} \tag{34}$$

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.

4.6. Fix-and-optimize heuristic as restart strategy

During the search process, if $f(v)$ 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 v. 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 v are fixed and the mission list $\{\mathcal{M}_{rt}, r \in \mathcal{R}, t \in \mathcal{T}\}$ of v 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 v' with an updated mission list $\{\mathcal{M}^{new}_{rt}, r \in \mathcal{R}, t \in \mathcal{T}\}$ and objective value $f(v')$.
- **Case 2:** The binary decision variables pertaining to $\{\mathcal{M}_{rt}, r \in \mathcal{R}, t \in \mathcal{T}\}$ of v 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 v' with objective value $f(v')$ and new pick list $\{\mathcal{P}^{new}_k, k \in \mathcal{K}\}$.

To restart the algorithm, one of the two cases is randomly selected to replace the current solution v and the corresponding objective value with v' and $f(v')$, 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 $f(v)$. Finally, leveraging optimization model is likely to direct the restart point to more promising solution regions.

4.7. 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 3. 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 (v^*) 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 3 Variable Neighborhood Search

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

5. Numerical experiments

In this section, we validate the proposed MILP model for CHR-OBASRP for small problem instances and also leverage the optimal solution to evaluate the performance of the metaheuristic approaches, namely VND, SA-ANS, and RSA-ANS. In addition, extensive numerical analysis is conducted to (i) assess the impact of adopting the fix-and-optimize restart strategy, (ii) benchmark the solution quality of RSA-ANS against VND since it was commonly used in prior works for efficiently solving OBASRP, (iii) gain insights on impact of human–robot composition, AMR speed, AMR cart capacity, and warehouse layout on total tardiness. To conduct the analysis, we generate test instances as described in Section 5.1. The parameter tuning procedure for the metaheuristic approaches, and the results are presented in Sections 5.2–5.6. The proposed MILP model is coded using Gurobi Python interface and the metaheuristic approaches are developed using Python 3.9. All experiments are executed on a desktop with an Intel Core-i9 processor and 128 GB RAM.

5.1. Generation of problem instances

To generate realistic test instances, we set the warehouse attributes based on the prior literature (Lee and Murray, 2019). Specifically, a single-block warehouse with 20 storage racks is considered, where each rack is 20 ft long and each storage location is 1-foot wide by 5 ft deep (Lee and 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. Likewise, the total items are fixed to 50 or 100 for generating larger instances. Also, our experimentation explores four different levels of human–robot team compositions $(k_{\max}, r_{\max})$ for collaborative order picking - (1,1), (2,1), (1,2) and (2,2) for small instances, and (2,2), (4,2), (2,4) and (4,4) for larger 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 and Murray, 2019). These speeds and warehouse configuration are used to establish the travel time of the

Table 3
Parameters and explored levels for SA-ANS and RSA-ANS.

Parameter	Values explored
$\theta^{\min}$	10, 20, 30
α	0.90, 0.95, 0.99
$\rho^{\max}$	$10 \cdot \mathcal{N}, 15 \cdot \mathcal{N}, 20 \cdot \mathcal{N}$
ρ^{RS} (applies only to RSA-ANS)	5, 10, 15
π	3, 4, 5

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 s, and the time taken to place it on the AMR is also 0.75 s (Lee and 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 ($k_{\max}$), and the modified traffic congestion rate (γ), a parameter that affects due date tightness (Lee and Murray, 2019). Given these four factors, the due date associated with order $j \in \mathcal{J}$ is randomly generated from the interval $\left[\hat{C}_j, (2 \cdot (1-\gamma) \sum_{j \in \mathcal{J}} \hat{C}_j + \min_{j \in \mathcal{J}} \hat{C}_j)/\min\{k_{\max}, r_{\max}\}\right]$. 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 research, a problem class is established based on three parameters, namely, $\mathcal{N}$, $(k_{\max}, r_{\max})$, and γ. Based on the above described values of these three parameters, we generate 12 different problem classes for both small ($|\mathcal{N}| = 10$ and 15) and large cases ($|\mathcal{N}| = 50$ and 100). Besides, we generate 10 instances for each problem class, thereby resulting in a total of $12 \cdot 10 \cdot 4 = 480$ instances. In addition, the large problem settings (240 instances) are also considered to assess the impact of AMR capacity, AMR speed, and multi-block warehouse layouts.

5.2. Parameter setting for metaheuristic approaches under study

The solution quality of SA-ANS and RSA-ANS depends on their hyperparameters. We established the initial temperature (θ^0) based on the slightly modified metropolis criteria that use the relative difference between the current and neighboring objective values. Specifically, we set θ^0 as 475 since it allows the algorithm to accept solutions that are 50% worse than current solution with a probability of 0.9 (i.e., $\theta^0 = -50/\log_e(0.9)$ based on metropolis criteria) (Parthasarathy and Rajendran, 1997). Furthermore, the epoch length ($I^{\max}$) is fixed to 30 since it is considered a statistically good sample (Parthasarathy and Rajendran, 1997). To fix the other key parameters, we adopt the procedure described in Vincent et al. (2021). Given a set of potential values for each parameter, their procedure evaluates different combinations of these values and selects the setting that achieves the best performance for a representative set of sample instances. In particular, we sequentially vary the value of a specific parameter while fixing the other to assess the impact of that parameter. The parameter values explored for SA-ANS and RSA-ANS are based on an initial empirical analysis and are given in Table 3. The parameters that produced the best performance values are as follows: $\theta^{\min} = 20, \alpha = 0.95, \rho^{\max} = 10 \cdot |\mathcal{N}|, \rho^{RS} = 5$ and $\pi = 3$. On the other hand, the VND algorithm does not contain parameters that require tuning.

5.3. Results of exact and metaheuristic approach applied to small instances

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 three metaheuristic approaches are also used for solving all the small instances, and their performance is compared against the optimal solution. Table 4 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. 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 s.

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 $k_{\max}$ is increased. For example, when $(k_{\max}, r_{\max})$ 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 (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).

The results in Table 4 also 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 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 s.

To evaluate the performance of the MILP solver for relatively large instances, we solved it for instances involving 50 items. Fig. 9 compares the solution obtained by the MILP model (after 7200 s) with the three metaheuristic approaches. The time taken by VND, SA-ANS, and RSA-ANS to obtain the least tardiness values is less than 1800 s. 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 $(k_{\max}, r_{\max}) = (2,2)$) to 91% ($\gamma = 0.6$ and $(k_{\max}, r_{\max}) = (4,2)$). 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.

Table 4
Performance comparison of metaheuristic approaches against the exact approach.

$\mathcal{N}$	(k_{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)

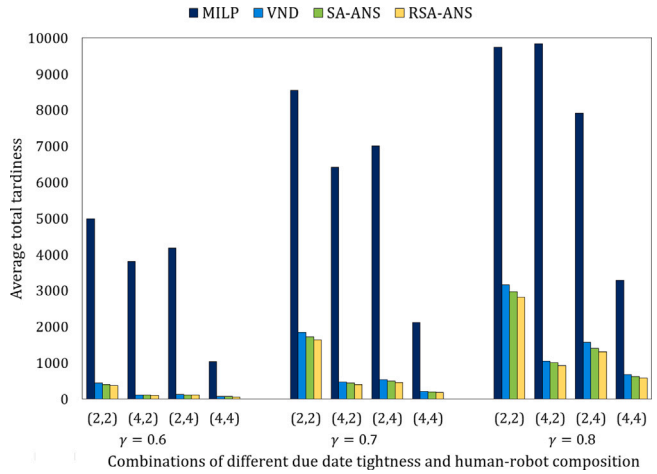

Fig. 9. Impact of varying AMR cart capacity on average total tardiness for instances with $|\mathcal{N}| = 50$.

5.4. 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. 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 $\Delta_{SA\text{-}ANS}$ 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 $\Delta_{SA\text{-}ANS}$, the average total tardiness obtained without the proposed restart mechanism (i.e., SA-ANS) is 6.86% and 7.70% higher for problem instances with $|\mathcal{N} = 50|$ and 100, respectively. For different values of both (k_{max}, r_{max}) and γ, the total tardiness obtained by RSA-ANS is consistently between 7%–8% lower than SA-ANS.

5.5. 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. Figs. 10(a) and 10(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 (k_{max}, r_{max}) is (2,2). It is about 85% and 90% for instances involving 50 and 100 items, respectively. Nevertheless, if (k_{max}, r_{max}) is set as (4,2)

Table 5
Average total tardiness for large instances, and solution quality of proposed RSA-ANS.

$\mathcal{N}$	(k_{max}, r_{max})	γ	VND	SA-ANS	RSA-ANS	Δ_{VND}	$\Delta_{SA\text{-}ANS}$
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%

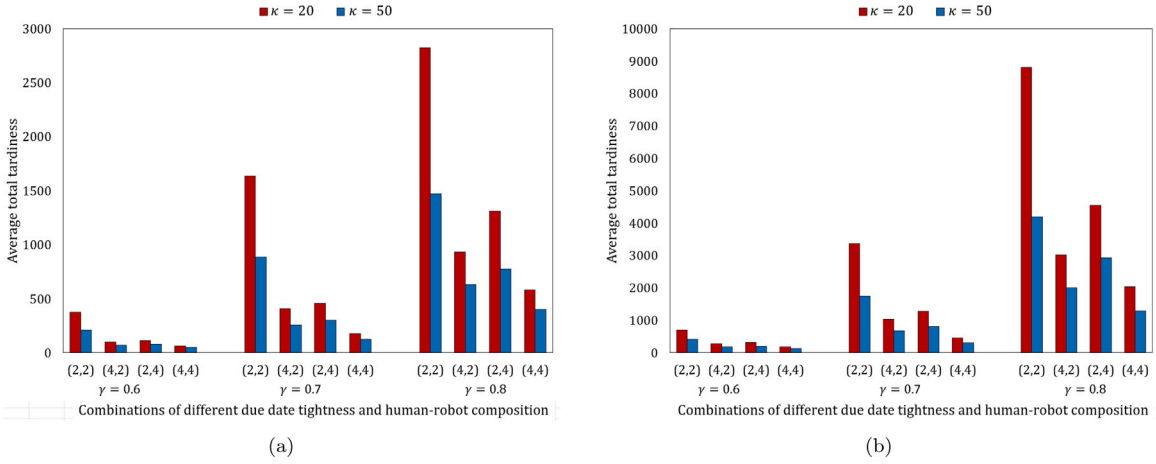

Fig. 10. Impact of varying AMR cart capacity on average total tardiness for instances with (a) 50 and (b) 100 items.

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, 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.

5.6. Impact of AMR speed

The initial analysis considered the AMR travel speed to be 2 ft/s according to related prior literature (Lee and 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/s from its baseline value.

Fig. 11(a) illustrates the percentage increase in the average total tardiness for all large problem classes when the AMR speed is decreased from 2 ft/s to 1 ft/s (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

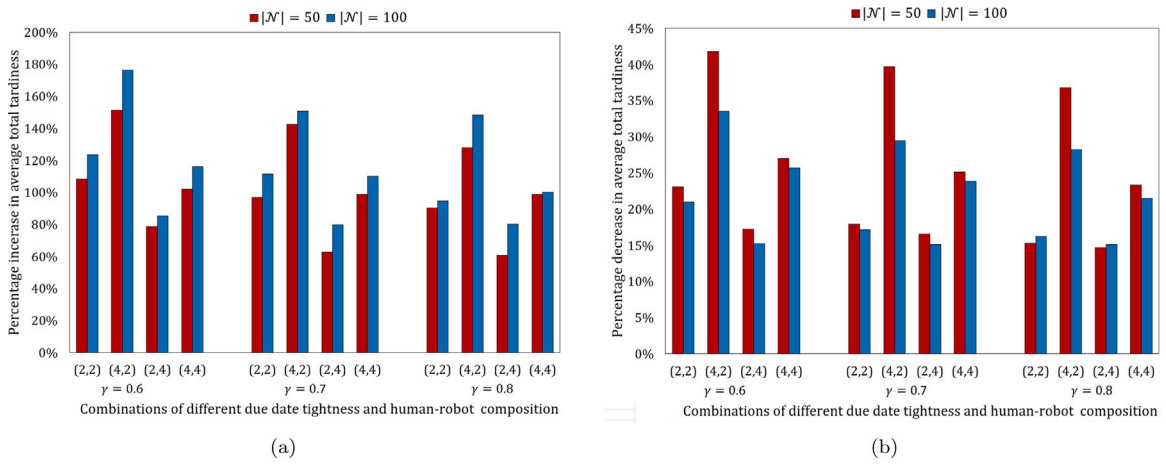

Fig. 11. Impact of (a) decreasing AMR travel speed from 2 ft/s to 1 ft/s and (b) increasing AMR travel speed from 2 ft/s to 3 ft/sec.

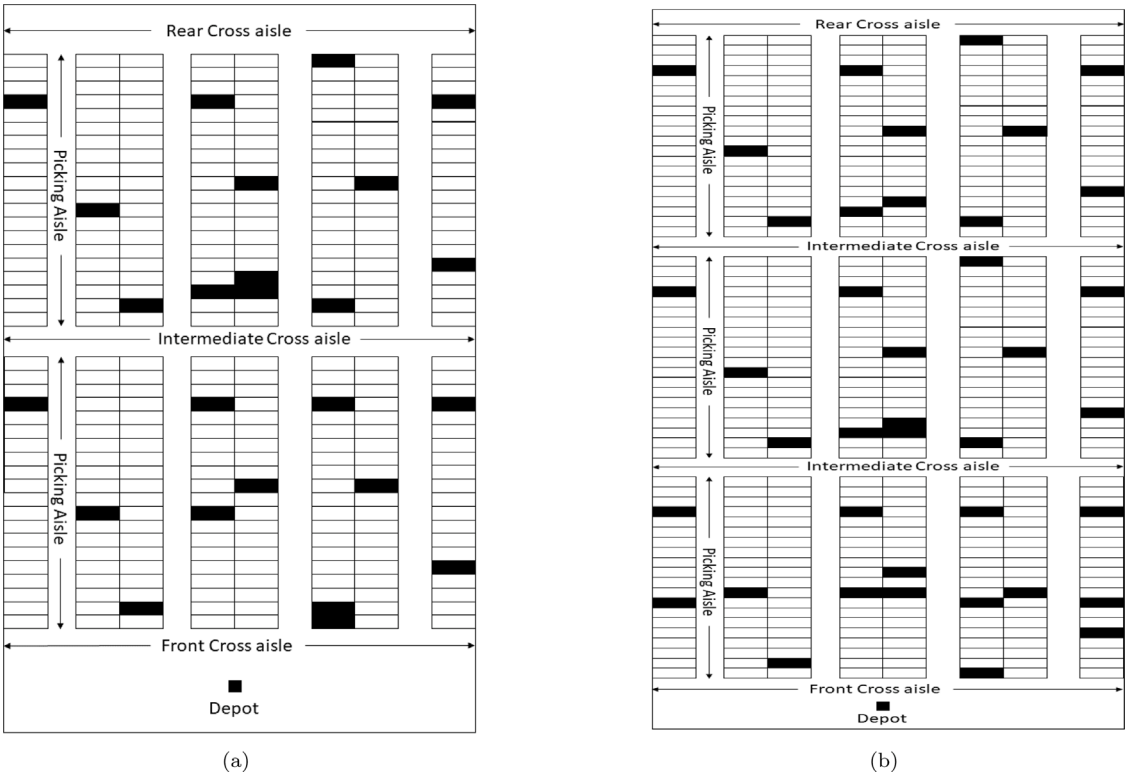

Fig. 12. Illustration of (a) two-block and (b) three-block warehouse layouts.

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 $(k_{\max}, r_{\max})$ 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.

Fig. 11(b) shows the impact of increasing AMR speed from 2 ft/s 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), 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 (Fig. 11) 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.

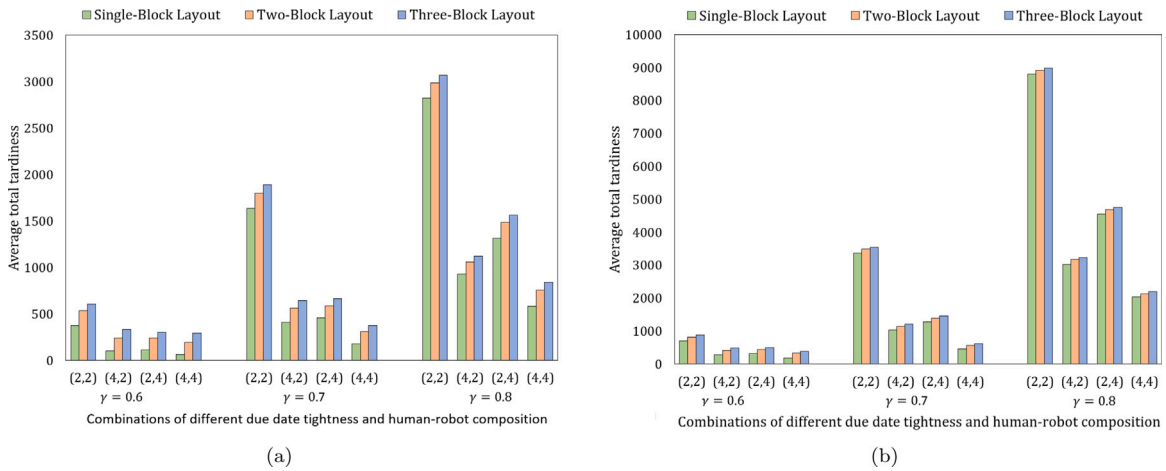

Fig. 13. Impact of different warehouse layouts on total tardiness for instances with (a) 50 and (b) 100 items.

5.7. 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 Fig. 12, an intermediate cross-aisle separates the blocks in multi-block warehouse layouts. Similar to the baseline setting discussed in Section 5.1, 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.

Figs. 13(a) and 13(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.

6. Conclusions

In this paper, we consider a collaborative order picking system (OPS), where human workers retrieve orders from warehouse storage and autonomous mobile robots (AMRs) transport them from picking area to the depot. Unlike existing works, this research considers the three subproblems associated with a multi-human, multi-robot OPS, and introduces the Collaborative Human–Robot Order Batching, batch Assignment and Sequencing, and picker-robot Routing Problem (CHR-OBASRP). A mixed integer linear programming model with the objective of minimizing the total tardiness is developed to solve the CHR-OBASRP to optimality. To deal with the computational intractability of the optimization model, we propose a restarted simulated annealing algorithm with adaptive neighborhood search (RSA-ANS), which integrates an adaptive multi-neighborhood search and fix-and-optimize 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 CHR-OBASRP, particularly the influence of AMR cart capacity, AMR speed and human–robot team composition.

For future work, the new variant on the collaborative order picking approach introduced in this research can be extended in numerous ways. First, we can integrate the fleet optimization decision into the CHR-OBASRP as our findings demonstrate human–robot team composition to have a substantial impact on total tardiness. Second, a comparative analysis of different warehouse layouts, pick strategies (e.g., zone picking vs. cluster picking) and human–robot routing strategies (e.g., picker-in lead vs. swarm) can be conducted to assess their influence. Another promising direction is to consider a dynamic system in which the order requests are arriving sequentially over time, thereby requiring real-time adjustments to the decisions associated with CHR-OBASRP. Finally, future research can consider other objectives (e.g., minimizing picker's travel distance) as well as trade-offs among multiple conflicting objectives (e.g., minimizing operating cost vs. makespan).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

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. Appl. Soft Comput. 75, 106–119. http://dx.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. Int. J. Prod. Econ. 206, 169–183. http://dx.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 Electron. J. http://dx.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. Int. J. Prod. Res. 1–20.

Caputo, A.C., Pelagagge, P.M., 2006. Management criteria of automated order picking systems in high-rotation high-volume distribution centers. Ind. Manag. Data Syst. 106, 1359–1383. http://dx.doi.org/10.1108/02635570610712627.

Cergibozan, Çağla., Tasan, A.S., 2019. Order batching operations: an overview of classification, solution techniques, and future research. J. Intell. Manuf. 30, 335–349. http://dx.doi.org/10.1007/S10845-016-1248-4.

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. Int. J. Prod. Econ. 159, 158–167. http://dx.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. Int. J. Prod. Econ. 170, 805–814.

de Koster, R., Le-Duc, T., Roodbergen, K.J., 2007. Design and control of warehouse order picking: A literature review. European J. Oper. Res. 182, 481–501. http://dx.doi.org/10.1016/j.ejor.2006.07.009.

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

Fager, P., Sgarbossa, F., Calzavara, M., 2021. Cost modelling of onboard cobot-supported item sorting in a picking system. Int. J. Prod. Res. 59, 3269–3284. http://dx.doi.org/10.1080/00207543.2020.1854484.

Ferreira, C., Figueira, G., Amorim, P., 2021. Scheduling human-robot teams in collaborative working cells. Int. J. Prod. Econ. 235, http://dx.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 Trans. (Inst. Ind. Eng.) 37, 63–75. http://dx.doi.org/10.1080/07408170590516917.

Goetschalckx, M., Ashayer, J., 1989. Classification and design of order picking. Logist. World 2, 99–106. http://dx.doi.org/10.1108/EB007469.

Henn, S., 2015. Order batching and sequencing for the minimization of the total tardiness in picker-to-part warehouses. Flex. Serv. Manuf. J. 27, 86–114. http://dx.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. Comput. Ind. Eng. http://dx.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. Int. J. Prod. Res. 4489–4505. http://dx.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. Int. J. Prod. Res. 57, 5821–5841. http://dx.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. J. Manuf. Syst. 54, 24–34. http://dx.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 J. Comput. http://dx.doi.org/10.1287/ijoc.2021.1060.

Napolitano, M., 2012. 2012 Warehouse/DC operations survey: mixed signals. Logist. Manag. (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 J. Oper. Res. 237, 1054–1066. http://dx.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. Prod. Plan. Control 8 (5), 475–483.

Rahman, S.M.M., Wang, Y., 2015. Dynamic affection-based motion control of a humanoid robot to collaborate with human in flexible assembly in manufacturing. In: ASME 2015 Dynamic Systems and Control Conference, Vol. 3. DSCC 2015, American Society of Mechanical Engineers, http://dx.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. Oper. Res. 31, 507–521. http://dx.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 J. Oper. Res. 263, 461–478. http://dx.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. CEJOR Cent. Eur. J. Oper. Res. 25 (2), 491–520.

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

Vincent, F.Y., Jewpanya, P., Redi, A.P., Tsao, Y.-C., 2021. Adaptive neighborhood simulated annealing for the heterogeneous fleet vehicle routing problem with multiple cross-docks. Comput. Oper. Res. 129, 105205.

Žulj, I., Salewski, H., Goeke, D., Schneider, M., 2021. Order batching and batch sequencing in an AMR-assisted picker-to-parts system. European J. Oper. Res. http://dx.doi.org/10.1016/j.ejor.2021.05.033.