IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 30, 2022, accepted April 12, 2022, date of publication April 21, 2022, date of current version April 29, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3169181

Batch Assorting for Worker-Following Assortment Carts in Parallel-Aisle Order-Assorting Systems

TAEHOON LEE, JEONGMAN LEE, AND SOONDO HONG, (Member, IEEE)
Department of Industrial Engineering, Pusan National University, Busan 46241, South Korea

Corresponding author: Soondo Hong (soondo.hong@pusan.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT of the Korea government (MSIT) under Grant NRF-2020R1A2C2004320 and by the BK21 FOUR of the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No.5199990914451).

ABSTRACT This study introduces an order-assorting system (OAS) in a distribution center. The system supports assortments with worker-following carts. The workers and worker-following carts move during an order-assortment operation before which the binning operation splits the large-volume stock-keeping units (SKUs) into bins according to the number of aisles. We propose two mixed-integer programming models. The batching-only model (BOM) conducts the batching operation to shorten the total travel distance. The binning and batching model (BBM) assumes that all SKUs are split into bins according to the number of aisles and finds the optimal point between binning and batching. We also propose the route packing-based binning then batching (RPBB) heuristic to solve a large-sized BBM problem. RPBB consists of a binning procedure based on route packing (BPM-RP) and a batching procedure using a simple integer programming formulation. The results of the experiments evaluating the performance of the BBM and the RPBB heuristic show that the model and heuristic optimize the balance between binning and batching to reduce the total travel distance. In the large-sized problem, the RPBB obtains near-optimal solutions by the tight lower bound that shows 1.41-2.30% optimal gaps on average.

INDEX TERMS Binning and batching, Heuristic algorithm, order-assorting operation, warehouse, worker-following cart.

I. INTRODUCTION

Consumers today are demanding fresher stock-keeping units (SKUs) and greater shopping convenience. Since retailers, especially convenience stores in urban areas, tend to keep minimal inventories due to the short life cycles of SKUs and the lack of storage space [1], their supply chains have shifted toward supplying fresh SKUs and minimum inventory. In the past, manufacturers generally supplied SKUs to individual stores, but most stores now supply SKUs from their own distribution centers (DCs) [2].

A retail convenience store's order fulfillment center (OFC), a type of DC, uses order-assorting (OA) to distribute the requested SKUs. The OFC supplies SKUs more than twice a day on average, considering the freshness of the SKUs ordered and the amount of storage space. The OFC supplies the sales volume of each SKU at stores and the fluctuations in the order sizes determine the daily supply operation.

The OFC needs to classify SKUs quickly and supply them on time. Automated OFCs handle large assortments, small orders, daily deliveries, and multiple types of workloads [3]. In this study, we consider a parallel-aisle order-assorting system (OAS) based on the worker-following assortment carts which load SKUs from a depot and unload them at the convenience stores' designated cells in the OAS. Each cart shows the cell locations, number of SKUs and their distributions, and other data. An example of an OAS is shown in Figure 1.

In general, batch assorting is popular in the OAS. To minimize the travel distance, the OAS combines the SKUs that require distribution into one trip (batching) or divides them into multiple trips (binning). This study makes two contributions to the binning and batching literature. First, we formulate a mixed-integer programming (MIP) model for binning and batching operations. The model minimizes the

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Gaggero.

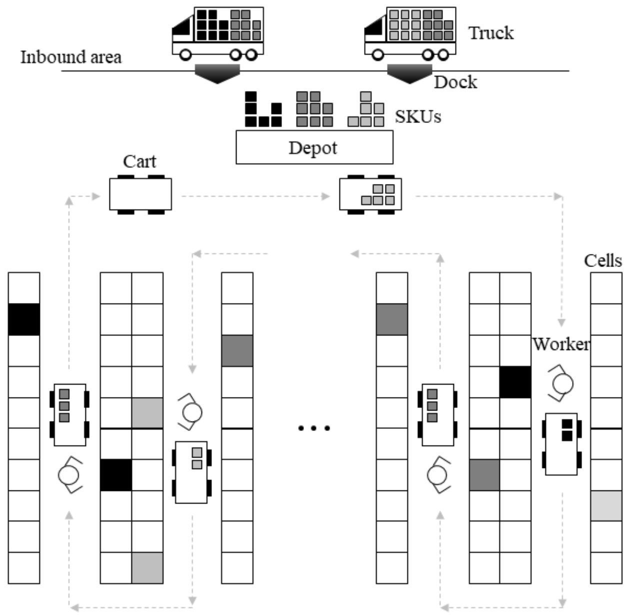


FIGURE 1. A parallel-aisle order-assorting system with worker-following carts (modified from Lee *et al.* [4]).

total travel distance of a cart by obtaining optimal bins and batches for small-sized problems. Second, we propose the route packing-based heuristic to solve a practical situation in large-sized problems with the binning and batching operations in the OAS.

The remainder of this study is organized as follows. Section II reviews the literature on the OAS, autonomous order-picking systems, and batch operations. Section III explains the binning and batching problems and batch assorting in OAS. Section IV introduces the MIP model. Section V describes the route packing-based binning then batching (RPBB) heuristic. Section VI evaluates the proposed model and heuristic via computational experiments. Section VII summarizes the contributions of this study and suggests possible directions for future research.

II. LITERATURE REVIEW

A. ORDER-ASSORTING SYSTEMS

Several summaries of the literature on order processing at DCs have been published [5]–[8]. Hong [1] introduced a worker-to-part OAS and evaluated the mean value and variance of worker's process time including sorting times, walking times, empty walking times, and blocking delays in a two-worker collaboration situation. Lee *et al.* [4] introduced an order batching procedure considering optimal travel distance Tan *et al.* [9], who studied parcel sorting in a warehouse using automated guided vehicles (AGVs), presented a mixed-integer linear programming model and developed a particle swarm optimization algorithm to minimize the completion time for the allocation of parcels, pick stations, and AGVs.

Many studies mainly focused on sortation conveyor systems or cross-docking. Boysen *et al.* [10], who reviewed automated conveyor systems for sortation from the perspective of operational research, used a layout design that considered multiple inbound and outbound stations. Fedtke and Boysen [11] introduced design alternatives for closed-loop tilt tray sortation conveyors in a parcel DC, formulated subproblems for system performance evaluation, and conducted simulations to compare system performance. Johnson and Meller [12] developed analytical models to evaluate the performance of a circular sorting conveyor system that sorted orders from a customer or retail store.

Cross-docking is associated with assorting operations Agustina *et al.* [13] proposed an integrated vehicle routing and scheduling model for a food DC using cross-docking. Enderer *et al.* [14] presented two MIP models and developed a column generation algorithm that minimized the total cost of material handling and transportation for an integrated cross-dock door assignment and the related vehicle routing problem. Yu *et al.* [15] studied the vehicle routing problem between the inbound and outbound routes related to cross-docking and proposed a simulated annealing-based heuristic algorithm. Nassief *et al.* [16] presented two MIP formulations for the dock-door assignment problem and proposed a column generation algorithm. Molavi *et al.* [17] developed a MIP model and four meta-heuristic algorithms for inbound and outbound truck scheduling in cross-docking systems with fixed due dates and shipment sorting.

B. AUTONOMOUS ORDER-PICKING SYSTEMS

Increasingly, DCs are implementing order-picking and storage technologies [18]. Worker-following cart systems are especially suitable for e-commerce DCs with strong demand fluctuations and large inventories of small SKUs.

Foumani *et al.* [19] considered an automated storage and retrieval system (ASRS). They developed a mixed-integer linear programming model to provide the optimal solution for robot moving sequences in small-sized problems, as well as a metaheuristic to solve large-sized problems efficiently. Kim and Hong [20] proposed two models for storage location assignment and reassignment in a bypass zone picking system with ASRS that took into account workload balancing between zones and recirculation reduction into account Boysen *et al.* [21] noted that in a rack-moving mobile robot environment, the mobile robot system transferred racks near picking stations; the optimized order processing could reduce the fleet size of robots by 50% or more.

Lamballais *et al.* [22] developed analytical models to evaluate the performance and utilization of robots in a robot mobile fulfillment system (RMFS). They confirmed the effect of the location of the workstation on the system's maximum order throughput. Kim *et al.* [23] developed a heuristic algorithm to solve an item assignment problem in the RMFS. Zou *et al.* [24] built a performance estimation model for the battery management problem in an RMFS, considering battery switching and charging strategies. They suggested a decomposition method for solving and validating the analytical models via simulation. Bolu and Korcak [25] proposed an adaptive heuristic approach for centralized task management in an RMFS. They performed simulations in a highly realistic environment including robot charging, replenishment process, and path planning algorithms to evaluate the proposed algorithms. Roy *et al.* [26] developed analytical models to evaluate the system operations for both single and multiple storage zones with dedicated or pooled robots in a mobile fulfillment system. Gharehgozli and Zaerpour [27] considered a scheduling problem with the objective of minimizing the total travel time of a mobile robot in RMFS. They developed an adaptive large neighborhood search algorithm and validated it by obtaining near-optimal solutions to the problem.

C. BATCHING OPERATION

Several studies have investigated batching operations to improve system efficiency. The batching algorithms generally fall into two categories: exact solution approaches and heuristic approaches.

Branch-and-bound algorithm [28], branch-and-price algorithm [29], and a column generation algorithm [30] are examples of exact solution approaches. Gademann *et al.* [31] proposed the branch-and-bound algorithm for the batching operation in a parallel-aisle warehouse. Gademann and Velde [32] proposed a branch-and-price algorithm to minimize the total travel time for the batching operation. Muter and Oncan [33] developed an order batching algorithm based on column generation considering traversal, return, and midpoint routing policies. Muter and Oncan [34] proposed a column generation-based algorithm to minimize a makespan objective for the integrated order batching and picker scheduling problem.

Heuristic approaches have introduced the first-come-first-served rule [35], seed algorithm [36], and saving algorithm [36], [37]. Hong *et al.* [38] introduced a route-packing-based order batching procedure (RBP) for large-scale problems that transformed the order batching problem into a route-bin packing problem (RPP). Similarly, Hong and Kim [39] developed an RBP for the S-shape routing policy in a parallel-aisle warehouse Hsu *et al.* [40] proposed a metaheuristic based on a genetic algorithm to minimize the total travel distance for solving medium- and large-sized order batching problems. Pan *et al.* [41] proposed a metaheuristic based on a group genetic algorithm for balancing the workload of each picking zone and minimizing the number of batches in a pick-and-pass system to reduce the total operation time. Matusiak *et al.* [42] used simulated annealing to solve an order batching problem with precedence constraints. Kulak *et al.* [43] used a tabu search to solve the order batching and picker routing problem, and Li *et al.* [44] used ant colony optimization to solve it.

We focus on the binning and batching problem in the OAS. We believe that binning during the batching operation has been addressed in the available literature. This study aims to optimize the batch assorting problem concerning binning in the OAS. Our optimization objective for the binning and batching problem is to minimize the total travel distance for a cart in the OAS. For the small-sized problems, we propose two formulations: a batching only model (BOM) and a binning and batching model (BBM). The effect of binning is demonstrated by comparing the results obtained from the two formulations. In addition, we propose a heuristic for large-sized problems concerning binning and batching problems. A comparison of the heuristic's results to those of a lower bound model demonstrates that the heuristic provides a near-optimal solution in large-sized problems.

III. PROBLEM DEFINITION

A. ORDER-ASSORTING IN A PARALLEL-AISLE OAS

Our study considers the order-assortment process in a parallel-aisle OAS where the SKUs arrive in bulk unit lots and that the DC uses worker-following carts. If there is little customer demand for an SKU it arrives in a small volume and is batched without splitting into smaller bins. If there is significant customer demand for the SKU, its large volume is first split into bins and is then batched. The batch assorting operation for SKUs uses a one-way traversal routing policy [8] as shown in Figure 2. A cart loads SKUs from the loading depot and travels to the cells assigned to the order. The cart in the OAS visits all the aisles to distribute the high-demand SKUs, and fewer aisles to distribute the low-demand SKUs. Assuming variable order sizes consisting of small-sized orders, the OAS uses a discrete batch assorting operation to combine multiple SKUs in one trip as shown in Figure 2 (a).

In an order picking operation, the DC splits and packs orders into a single order after completion of each sub-order retrieval, or delivers the packed shares separately to

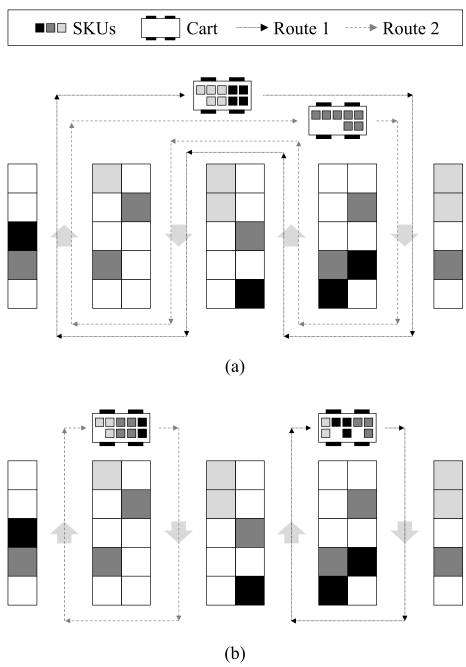


FIGURE 2. Batch assorting operation using the one-way traversal routing policy: (a) without binning and (b) with binning.

the customer's assigned bin. Although discrete assorting and order picking can both increase the operational time and picking cost, binning in the assorting operation becomes a partial distribution without any added cost. We aggregate multiple SKUs in one trip (batch assorting) in order to reduce the overall total travel distance, assuming order size variability and collision-free carts in the OAS.

B. BINNING AND BATCHING PROBLEMS

The batch assorting operation has two problems, binning and batching, to consider when distributing SKUs into the boxes to be delivered to the customers. Binning refers to splitting the SKUs according to each aisle. If the DC does not use assorting carts, and the volume of the SKUs is small, it is necessary to split the SKUs into a number of bins equal to the number of aisles; but if it does use the carts, the only consideration for binning is the carts' volume capacity. Batching refers to grouping or clustering the bins into batches. Grouping the bins into batches to match the carts' volume capacity will reduce the number of trips and total travel distance as shown in Figure 2 (b). Since binning is interpreted as a set partitioning problem, the complexity of binning is NP-Complete [45]. Batching was proven to be NP-hard when the capacity of a batch was larger than three [32].

IV. BATCH ASSORTING FORMULATION

We develop two mathematical models: a batching-only model (BOM) and a binning and batching model (BBM) for the batch assorting. The aim is to form batches that can take the shortest routes. The constraints of the two models are of three types: (i) batching constraints, (ii) cart capacity constraints, and (iii) route constraints. Batching constraints ensure that at least one SKU is assigned to each batch, and cart capacity constraints ensure that the total volume of all the SKUs in each batch does not exceed the cart's maximum capacity. For simplicity, we consider only the volume of SKUs and neglect their shape of SKUs in satisfying cart capacity constraints. Route constraints ensure that a batch holds all of all SKUs in a route. *CAPA* represents the volume capacity of the carts. Route information is expressed by the aisle incidence (RA_{ra}), and the route length is LT_r. The parameters related to the SKUs are the list of SKUs, volume information of each SKU, and information on the aisles that the SKUs should visit. Let s be an SKU in a set of SKUs S. Let b be a batch in a set of batches B. In the BOM, an SKU is assigned to a batch. In the BBM, we allow the possibility for each SKU to be grouped into multiple batches. The volume of the SKU s that must be sorted in aisle a, V_{sa}, is measured in liters. PA_{sa} is the set of aisles that must be visited to sort the SKU s. To obtain a feasible solution, we assume the cart capacity must at least meet the following relationship: $\sum_{a\in\hat{o}A} V_{sa} \leq CAPA, \forall_s \in \hat{o}S$. The notations for the batch assorting formulation are summarized in Table 1.

TABLE 1. Notations for the batch assorting formulation.

Notation	Explanation
S	Set of SKUs (index $s \in S$)
A	Set of aisles (index $a \in A$)
B	Set of batches (index $b \in B$)
R	Set of routes (index $r \in R$)
LT_r	Length of route r
V_{sa}	Volume of SKU s assigned to aisle a
$CAPA$	Volume capacity of a cart
RA_{ra}	Binary variables: 1, if route r passes through aisle a; 0, otherwise (see Hong et al. [38] for details)
PA_{sa}	Binary variables: 1, if SKU s passes through aisle a; 0, otherwise

TABLE 2. Decision variables for the BOM.

Decision variable	Explanation
X_{sb}	Binary variables: 1, if SKU s's bin is assigned to batch b; 0, otherwise
Y_{br}	Binary variables: 1, if batch b takes route r; 0, otherwise
Z_b	Binary variables: 1, if at least one SKU's bin is assigned to batch b; 0, otherwise

A. BATCHING-ONLY MODEL

The batch assorting model assigns the SKUs into bins to shorten the total travel distance. For simplicity, we consider that each SKU is assigned to exactly one batch and the volume of each SKU is contained within the cart capacity. We propose a mixed-integer programming (MIP) model for the BOM. The BOM is the mathematical model for batching the SKUs without splitting them into bins. The decision variables for the BOM are summarized in Table 2.

$$BOM : \min \sum_{b\in B} \sum_{r\in R} LT_r \cdot Y_{br}, \tag{1}$$

$$\text{subject to} \sum_{b\in B} X_{sb} = 1, \quad \forall s\in S, \tag{2}$$

$$X_{sb} \leq Z_b, \quad \forall s\in S, \forall b\in B, \tag{3}$$

$$\sum_{s\in S} \sum_{a\in A} V_{sa} \cdot X_{sb} \leq CAPA, \quad \forall b\in B, \tag{4}$$

$$\sum_{r\in R} Y_{br} \leq Z_b, \quad \forall b\in B, \tag{5}$$

$$X_{sb} \cdot PA_{sa} \leq \sum_{r\in R} RA_{ra} \cdot Y_{br}, \quad \forall s\in S, \forall a\in A, \forall b\in B, \tag{6}$$

$$X_{sb} \in \{0, 1\} \quad \forall s\in S, \forall b\in B,$$

$$Y_{br} \in \{0, 1\}, \quad \forall b\in B, \forall r\in R,$$

$$Z_b \in \{0, 1\}, \quad \forall b\in B.$$

Objective function (1) minimizes the total travel distance that is, the sum of the length of the assigned route. We obtain

TABLE 3. Decision variables for the BBM.

Decision variable	Explanation
X_{sab}	Binary variables: 1, if SKU s's bin that visits aisle a is assigned to batch b; 0, otherwise
Y_{br}	Binary variables: 1, if batch b takes route r; 0, otherwise
Z_b	Binary variables: 1, if at least one SKU's bin is assigned to batch b; 0, otherwise

the appropriate route r for each batch. Constraint (2) assigns one SKU to each batch for the batch assorting without binning operation. An SKU cannot be separated into multiple batches. Constraint (3) validates a batch if a bin is assigned to the corresponding batch. Constraint (4) ensures that the total number of SKUs in one batch does not exceed the cart's capacity. Constraint (5) ensures that each batch takes a single type of route, and Constraint (6) ensures that the route assigned to a batch holds all the SKUs in the corresponding batch. The possible maximum number of batches that could be constructed in the BOM is the number of SKUs ($|S|$).

B. BINNING AND BATCHING MODEL

The binning and batching model splits all the SKUs into bins to shorten the total travel distance; for simplicity, we consider the volume of the SKU in the cart capacity. We propose a mixed-integer programming (MIP) for the BBM. We allow for the possibility that each SKU can be assigned to a few batches. The parameters related to the SKUs are the list of SKUs, volume information of each SKU, and information on the aisles that the SKUs should visit. The decision variables for the BBM are summarized in Table 3.

$$BBM : \min \sum\nolimits_{b\in B} \sum\nolimits_{r\in R} LT_r \cdot Y_{br}, \tag{7}$$

$$\text{subject to} \sum\nolimits_{b\in B} X_{sab} \geq 1, \quad \forall s\in S, \forall a\in A, \tag{8}$$

$$X_{sab} \leq Z_b, \quad \forall s\in S, \forall a\in A, \forall b\in B, \tag{9}$$

$$\sum\nolimits_{s\in S} \sum\nolimits_{a\in A} V_{sa} \cdot X_{sab} \leq CAPA, \quad \forall b\in B, \tag{10}$$

$$\sum\nolimits_{r\in R} Y_{br} \leq Z_b, \quad \forall b\in B, \tag{11}$$

$$X_{sab} \cdot PA_{sa} \leq \sum\nolimits_{r\in R} RA_{ra} \cdot Y_{br}, \quad \forall s\in S, \forall a\in A, \forall b\in B, \tag{12}$$

$$X_{sab}\in \{0, 1\}, \quad \forall s\in S, \forall a\in A, \forall b\in B,$$

$$Y_{br}\in \{0, 1\}, \quad \forall b\in B, \forall r\in R,$$

$$Z_b\in \{0, 1\}, \quad \forall b\in B.$$

Objective function (7) minimizes the total travel distance that is the sum of the length of the assigned route. Constraint (8) calculates the number of batches required. The BBM considers the binning problem in batch assorting. An SKU can be separated into multiple batches. The binning of SKUs considers the route including the aisles, which are the location of stores that find the SKUs. If the items in aisle a in an SKU s are included in batch b, SKU s should be filled by batch b. Items in an aisle are not split into multiple batches. Constraint (9) assigns items in at least one aisle to one batch. Constraint (10) ensures that a batch does not exceed the capacity of the carts. The equation sums the volume of items per aisle of the SKUs in a batch, then compares it with the cart capacity. Constraint (11) ensures that each batch takes only one route. Constraint (12) ensures that the route assigned to a batch covers all bins or parts of the SKUs in the corresponding batch.

V. THE ROUTE PACKING-BASED BINNING THEN BATCHING

This section describes the proposed heuristic algorithm for the large-sized problems. Due to the problem's NP-Hard and practical size, BBM is difficult to solve in a reasonable amount of time. The heuristic algorithm develops from BBM using the route packing-based binning then batching (RPBB) procedure. RPBB builds batches from the bins that hold the SKUs divided into bulk units. Simultaneously, it considers all the SKUs for each customer, splits the SKUs into bins, assigns the bins to batches, and performs the route selection.

RPBB consists of a binning and a batching procedure. In the binning procedure, the bulk units of SKUs are divided into the requirements for each route using a partitioning problem-based route packing (hereinafter, BBM-RP) model (Section 5.A). In the batching procedure, the SKUs that are divided by each route are assigned to batches, considering the capacity of the batches using a simple integer programming (IP) model (BP_r, Section 5.B). RPBB solves the BBM-RP model and BP_r model using an IP solver.

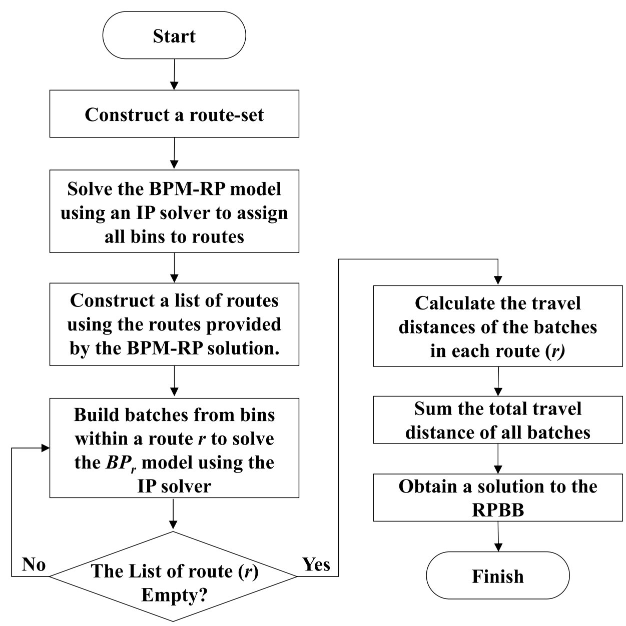


FIGURE 3. A flowchart of the RPBB.

TABLE 4. Decision variables for the BBM-RP.

Decision variable	Explanation
x_{sar}	Binary variables: 1, if SKU s's bin that should visit aisle a is assigned to route r; 0, otherwise
y_r	The number of SKUs assigned to route r

Figure 3 illustrates a flow chart for the relationship between the BBM-RP and BP_r models in RPBB.

A. BBM-RP MODEL FOR THE BINNING PROCEDURE

We simplify the BBM by removing the batching variables to develop the BBM-RP model. By skipping the batching stage, we relax the batching problem to assign the bins to routes and identify the number of routes required to assort the bins.

We reuse the two decision variables, X_{sab} and Y_{br} (introduced in Table 3), which we derive from the BBM. Using the following two equations, $x_{sar} = \sum_{b\in B} X_{sar} \times Y_{br}$ and $y_r = \sum_{b\in B} Y_{br}$, we define x_{sar} as the SKU s's bin that should visit aisle a if a is assigned to route r, and y_r as the number of SKUs assigned to route r. The decision variables for the BBM-RP are summarized in Table 4. The details and reformulation are as follows.

$$\mathrm{BBM-RP}: \min \sum_{r\in R} LT_r \cdot y_r, \tag{13}$$

$$\text{subject to} \sum_{r\in R} x_{sar} \geq 1, \quad \forall s\in S, \forall a\in A, \tag{14}$$

$$\sum_{s\in S} \sum_{a\in A} V_{sa} \cdot x_{sar} \leq CAPA \cdot y_r, \quad \forall r\in R, \tag{15}$$

$$x_{sar} \cdot PA_{sa} \leq RA_{ra} \cdot y_r, \quad \forall s\in S, \forall a\in A, \forall r\in R, \tag{16}$$

$$x_{sar} \in \{0, 1\}, \quad \forall s\in S, \forall a\in A, \forall r\in R,$$

$$y_r \in \{0, 1, \ldots, \}, \quad \forall r\in R.$$

Objective function (13) minimizes the total travel distance which is the sum of the lengths of all the routes assigned. Constraint (14) assigns all the bins to exactly one route, Constraint (15) ensures that the capacity of the assigned route should be greater than or equal to the total volume of bins to assort, and Constraint (16) ensures that the aisle incidence vector of route r should contain the aisle incidence vector of each bin s that has been assigned to route r.

Based on the two decision variables, $x_{sar,}$ and y_r, we derive Constraints (14), (15), and (16) using Gaussian elimination from the BBM. We match the constraints specified by bin s in Constraint (9) to the same constraints in constraint (14). Constraints (9) and (10) are valid after aggregating the constraints related to route r. We replace batching index b with route index r by aggregating the constraints with the same route index r. Therefore, the BBM-RP model has no batching index. Given route r, Constraint (15) denotes the number of routes required. We repeat the process for Constraint (12) to obtain Constraint (16), which ensures that route r can assort bin s by comparing the aisle indicator parameters of route r and the SKU s's bin that should visit aisle a. We skip the batching constraints to derive a relaxation model without the batching variables. The BBM-RP model, however, still contains the partitioning constraints (Constraint (14)).

TABLE 5. Decision variables for the BP_r.

Decision variable	Explanation
x_{sb}	Binary variables: 1, if pair s from bin-to-route assignment information is assigned to batch b; 0, otherwise
z_b	Binary variables: 1, if bins are assigned to batch b; 0, otherwise

B. BPR MODEL FOR THE BATCHING PROCEDURE

The BP_r model constructs batches with routes using the bins-to-route assignment information derived from the BPM-RP model. The BP_r in the RPBB merges bins into batches and determines the number of batches (z_b) per route. The maximum number of B_r is the one that assumes Max (the number of SKUs, y_r+ 3). We reuse the parameter, V_{sa} (introduced in Table 1), using the following equation, $\sum_{r\in R} \sum_{s\in S_r} V_s = \sum_{s\in S} \sum_{a\in A} V_{sa}$, we define V_s as the bin volume of pair s from bin-to-route assignment information. The decision variables for the BP_r are summarized in Table 5.

$$BP_r : \min \sum_{b\in B_r} z_b, \tag{17}$$

$$\text{subject to} \sum_{b\in B_r} x_{sb} = 1, \quad \forall s\in S_r, \tag{18}$$

$$\sum_{s\in S_r} V_s \cdot x_{sb} \leq CAPA \cdot z_b, \quad \forall b\in B_r, \tag{19}$$

$$x_{sb} \in \{0, 1\}, \quad \forall s\in S_r, \forall b\in B_r,$$

$$z_b \in \{0, 1\}, \quad \forall b\in B_r.$$

Objective function (17) minimizes the number of batches, Constraint (18) assigns one bin to each batch, and Constraint (19) ensures that the total volume of bins in a batch does not exceed the capacity of carts.

VI. EXPERIMENT

Studying the effectiveness of binning allows us to compare batch assorting via the *without binning model* (BOM, Section 4) with the *with binning model* (BBM, Section 4). To evaluate the performance of the heuristic algorithm, we use the small-sized problems to compare the heuristic algorithm with BBM and the large-sized problems to compare the heuristic algorithm with the lower bound (LB, Appendix A). We also test the heuristic algorithm against the performance for various problem sizes.

TABLE 6. **Experimental characteristics.**

Profile	Parameters
Number of SKUs ($\|S\|$)	10, 20, 30, 360, 540, 720
Volume of SKUs	Uniform (0.5, 5.0)
Routing policy	One-way traversal
Number of aisles ($\|A\|$)	4, 6, 8
Length of an aisle (LT)	15 m
Distance between two adjacent aisles	4 m
Capacity of a worker-following cart ($CAPA$)	6, 12, 24, 30, 36 liters

A. EXPERIMENTAL DESIGN

We generate orders using a random order generator. The parameters include SKU information, route method, layout information, and worker-following cart information. The SKU information includes the number of SKUs and volume data. We assume the sets of a number of SKUs to be 10, 20, 30, 360, and 720, and consider a unit-size volume of SKUs such as milk, yogurt, and other dairy SKUs. The volume follows a uniform distribution from 0.5 to 5.0. The layout information includes number of aisles, aisle length, and distance between two adjacent aisles. Table 6 lists the experimental characteristics. Each experiment is repeated 25 times using random instances.

We conduct all the experiments on a PC with Windows 10 running on Intel i5-8600K @ 3.60 GHz and 32.0 GB RAM. All the experiments comprise five random instances. We use IBM ILOG CPLEX version 12.6 to solve the MIP formulations. We implement the MIP models using Concert Technology in IBM Java.

We use the following notations:

- BOM: Exact solution of the batching only model (without binning)
- FFD: Shortest route with first-fit decreasing (Appendix B)
- RPBB: The route-set packing based binning then batching
- BBM: Exact solution of the binning and batching model
- LB: Lower bound for the evaluation of heuristics

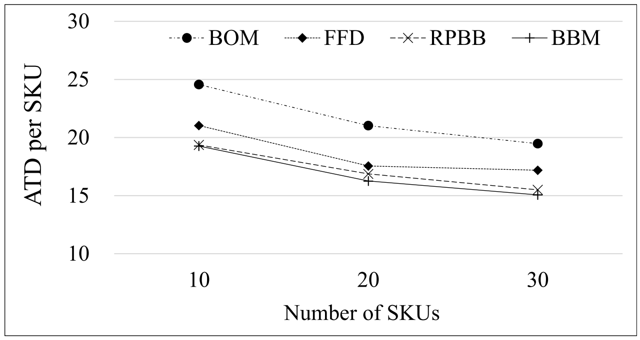


FIGURE 4. **Results of the average travel distance per SKU in small-sized problems.**

We consider three performance measures: total travel distance, number of batches, and computation time. The notations are as follows:

- # of batches: Number of batches (EA)
- Obj: Total travel distance (meters)
- Opt Gap: Objective gap between the BBM and heuristics:

$$\mathrm{OptGap}(\%) = 100 \times \left(\frac{\text{Obj.of the heuristics} - \text{Obj.of the Opt}}{\text{Obj.of the heuristics}}\right)$$

- LB Gap: Objective gap between the LB and heuristics:

$$\mathrm{LBGap}(\%) = 100 \times \left(\frac{\text{Obj.of the heuristics} - \text{Obj.of the LB}}{\text{Obj.of the heuristics}}\right)$$

- CPU: Computation time (seconds)

B. COMPUTATIONAL RESULTS FOR SMALL-SIZED PROBLEMS

We consider 10-, 20-, and 30-SKU cases in 4- and 6-aisle OAS. The results are reported in Table 7. The optimal model obtains the shortest total travel distance (this measure is the objective function of BBM) within 60 s for the 10-, 20-, and

TABLE 7. **Computational results for small-sized problems.**

$\|A\|$	*Capa*	$\|S\|$	# of batches				Total travel distance				Opt Gap (%)		CPU (sec)			
			BOM	FFD	RPBB	**BBM**	BOM	FFD	RPBB	**BBM**	FFD	**RPBB**	BOM	**FFD**	RPBB	BBM
4	6	10	6.96	7.44	6.64	**6.36**	393.52	348.64	313.52	**302.16**	13.33%	**3.62%**	0.07	**0.00**	0.04	0.06
		20	12.68	13.84	11.92	**11.16**	693.44	640.48	555.28	**528.16**	17.54%	**4.88%**	0.33	**0.00**	0.09	0.26
		30	18.96	20.88	17.40	**16.52**	1019.68	958.56	801.36	**765.28**	20.16%	**4.50%**	0.55	**0.00**	0.17	2.99
	12	10	3.16	3.44	3.16	**3.12**	192.00	166.56	163.52	**161.36**	3.12%	**1.32%**	0.08	**0.00**	0.03	0.08
		20	5.68	6.28	5.72	**5.56**	320.96	298.80	272.24	**264.24**	11.57%	**2.94%**	0.29	**0.00**	0.06	0.25
		30	8.20	9.16	8.28	**8.20**	451.12	427.44	386.32	**380.72**	10.93%	**1.45%**	0.98	**0.00**	0.06	3.99
6	6	10	6.76	7.32	6.40	**6.32**	452.80	397.20	360.40	**352.08**	11.36%	**2.31%**	0.07	**0.00**	0.05	0.06
		20	12.80	13.84	12.92	**11.60**	850.48	736.48	700.56	**636.80**	13.53%	**9.10%**	0.35	**0.00**	0.14	0.75
		30	19.32	20.96	18.92	**16.88**	1270.80	1120.96	1027.36	**915.68**	18.31%	**10.87%**	0.80	**0.00**	0.38	2.41
	12	10	3.24	3.60	3.36	**3.32**	245.68	210.32	193.68	**192.72**	8.37%	**0.50%**	0.10	**0.00**	0.03	0.07
		20	5.84	6.36	5.92	**5.76**	420.56	351.12	337.44	**325.12**	7.40%	**3.65%**	0.65	**0.00**	0.08	0.44
		30	8.36	9.44	8.56	**8.28**	584.32	515.52	465.20	**452.00**	12.32%	**2.84%**	10.56	**0.00**	0.19	8.41

TABLE 8. Computational results for large-sized problems.

$\|A\|$	*Capa*	$\|S\|$	# of batches		Total travel distance			LB Gap (%)		CPU (sec)	
			FFD	**RPBB**	FFD	**RPBB**	LB	FFD	**RPBB**	**FFD**	RPBB
4	24	360	50.93	**47.82**	2353.57	**2204.07**	2162.24	8.13%	**1.90%**	**0.01**	17.64
		540	76.04	**71.36**	3515.64	**3294.43**	3247.01	7.64%	**1.44%**	**0.01**	50.09
		720	101.93	**95.32**	4708.71	**4397.64**	4335.42	7.93%	**1.41%**	**0.03**	76.70
	30	360	40.00	**38.32**	1852.16	**1770.40**	1731.74	6.50%	**2.18%**	**0.01**	22.12
		540	59.96	**57.40**	2776.40	**2653.52**	2603.90	6.21%	**1.87%**	**0.01**	48.29
		720	79.92	**76.56**	3692.32	**3533.92**	3471.33	5.99%	**1.77%**	**0.03**	81.26
	36	360	33.32	**31.92**	1543.92	**1471.52**	1443.12	6.53%	**1.93%**	**0.00**	2.12
		540	49.80	**48.04**	2307.12	**2221.04**	2169.91	5.95%	**2.30%**	**0.01**	49.00
		720	66.58	**64.16**	3083.35	**2965.61**	2906.09	5.75%	**2.01%**	**0.02**	67.96
6	24	360	50.84	**47.72**	2751.12	**2569.20**	2523.58	8.27%	**1.78%**	**0.03**	37.58
		540	76.31	**72.07**	4129.03	**3887.31**	3814.11	7.63%	**1.88%**	**0.05**	77.33
		720	101.36	**95.40**	5489.44	**5158.64**	5079.66	7.46%	**1.53%**	**0.09**	90.99
	30	360	40.17	**38.30**	2174.87	**2066.07**	2019.43	7.15%	**2.26%**	**0.02**	28.67
		540	59.92	**57.32**	3249.12	**3097.20**	3033.91	6.62%	**2.04%**	**0.05**	65.40
		720	79.94	**76.81**	4336.63	**4153.38**	4068.76	6.18%	**2.04%**	**0.08**	97.62
	36	360	33.20	**31.76**	1808.16	**1715.04**	1682.39	6.96%	**1.90%**	**0.02**	11.80
		540	49.72	**47.78**	2701.81	**2583.19**	2534.13	6.21%	**1.90%**	**0.06**	40.87
		720	66.16	**63.68**	3590.56	**3443.20**	3386.44	5.68%	**1.65%**	**0.09**	64.64
8	24	360	50.68	**47.72**	3168.08	**2967.92**	2910.21	8.14%	**1.94%**	**0.17**	45.17
		540	75.36	**71.20**	4692.16	**4421.36**	4334.92	7.61%	**1.96%**	**0.35**	81.41
		720	100.69	**95.46**	6257.69	**5917.85**	5803.95	7.25%	**1.92%**	**0.55**	108.37
	30	360	40.03	**38.07**	2511.67	**2378.13**	2326.71	7.36%	**2.16%**	**0.17**	37.88
		540	59.72	**57.08**	3738.16	**3554.00**	3481.00	6.88%	**2.05%**	**0.35**	56.87
		720	79.70	**76.47**	4969.93	**4749.40**	4652.47	6.39%	**2.04%**	**0.56**	86.30
	36	360	33.39	**31.91**	2105.04	**1994.61**	1951.33	7.30%	**2.17%**	**0.17**	31.41
		540	49.70	**47.83**	3109.30	**2975.65**	2915.30	6.24%	**2.03%**	**0.36**	44.17
		720	66.17	**63.65**	4133.39	**3956.87**	3887.00	5.96%	**1.77%**	**0.54**	57.16

30-SKU cases in the 6-aisle OAS. BBM always guarantees shorter travel distances and fewer batches. The binning contribution is shown by comparing BBM and BOM in operational performance. The shorter the travel distance, the more likely that each cart will maintain its dedicated area, i.e., the lesser the congestion. Moreover, a decrease in the number of batches can save loading and unloading times. In the 4-aisle OAS, FFD obtains 3.12-20.16% optimal gap solutions within 0.01 s and RPBB obtains 1.32-4.88% optimal gap solutions within 0.02-0.20 s for the 10-, 20-, and 30-SKU cases. In the 6-aisle OAS, FFD obtains 7.40-18.31% optimal gap solutions within 0.01 s, and RPBB obtains 0.50-10.87% optimal gap solutions within 0.06-0.65 s for the 10-, 20-, and 30-SKU cases.

Figure 4 shows the average travel distance (ATD) per SKU to compare four results (BOM, FFD, RPBB, and BBM) for small-sized problems in an OAS with 6 aisles and 12 cart's capacity. BBM, FFD, and RPBB developed for the binning and batching operations show significant improvements in the ATD per SKU over BOM. The ATD per SKU of BBM is 21.56-22.69% shorter than that of BOM. Meanwhile, the ATD per SKU of RPBB and FFD are compared, it is confirmed that RPBB is 3.90-9.76% shorter than FFD.

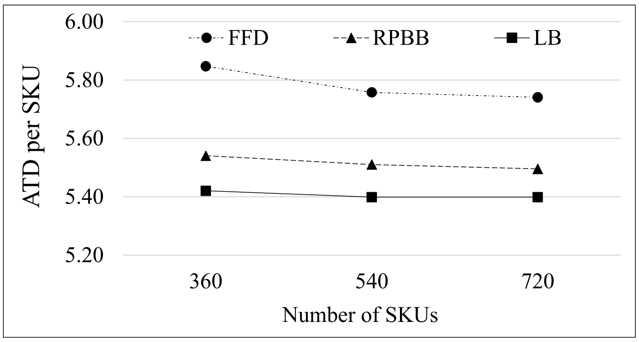


FIGURE 5. The results of the average travel distance per SKU in large-sized problems.

All the results in Figure 4 indicate that, as the number of SKUs increases from 10 to 30, the optimal gap decreases. Additionally, the optimal gap of RPBB is 0.50-3.65% compared to that of BBM, which is an optimal solution, indicating a slight difference.

C. COMPUTATIONAL RESULTS FOR LARGE-SIZED PROBLEMS

For the large-sized problems, we consider 360-, 540-, and 720-SKU cases with 4-, 6-, and 8-aisle OAS. In the

24-cart-capacity scenarios, the average number of batches created is 4.45, and in 30-cart-capacity scenarios, the average number of batches created is 2.56, and in 36-cart-capacity scenarios, the average number of batches created is 1.92 more in FFD than in RPBB.

We evaluate heuristic solutions in large-sized problems using the LB model. Because BBM was unable to obtain reasonable results in the problems within 3600 s. We follow Hong *et al.* [22] to avoid the computational burden when solving the problems and to derive the LB. The LB gap solutions between FFD and LB is approximately 5.68-8.27% as $|P|$ is 360, 540, and 720, respectively. Similarly, the LB gap solutions between RPBB and LB is about 1.41-2.30% as $|P|$ is 360, 540, and 720, respectively. The results of RPBB imply that RPBB consistently outperforms FFD and produces solutions within 2.30% of LB solutions. The results are reported in Table 8.

Figure 5 shows the ATD per SKU to compare the results of FFD, RPBB, and LB for large-sized problems in an OAS with 8 aisles and 36 carts' capacity. It shows that the ATD per SKU reduces when the number of SKUs increases from 360 to 720 for FFD, RPBB, and LB. The ATD per SKU of RPBB is 4.08-5.39% shorter than that of FFD. Additionally, the results of FFD and RPBB indicate that as the number of SKUs increases from 360 to 720, the optimal gap decreases. The LB gap solutions of RPBB is 1.77-2.17% compared to LB, indicating a small difference.

VII. CONCLUSION

In this study, we optimized the balance between binning and batching to shorten the overall total travel distance. Two MIP models were proposed. BOM describes assortment as a traditional batching model, which does not consider the cost of binning. BBM minimizes the total travel distance of carts for the OASs that use worker-following cart systems.

We propose BBM and RPBB that shorten the total travel distance by optimizing the balance between binning and batching. In the large-sized problem, RPBB obtains near-optimal solutions by the tight lower bound that shows 1.41-2.30% optimal gaps on average and can solve large-sized problems within 2 min.

Due to the economies of scale, very large DCs are replacing smaller DCs and warehouses [8]. The DCs' use of autonomous cart systems in assorting operations could manage strong fluctuations in demand at reduced cost. In particular, the performance of the RPBB could be beneficial for large DCs.

The proposed models and solutions in this study did not account for the blocking between carts. As a DC's square footage and number of carts increases, the impact of the blocking between carts on the operational performance is expected to increase. Research on a manual order picking system has considered the blocking between workers [38]. Future research should evaluate the congestion with an analysis of bottlenecks in multi-cart operations and develop optimal binning and batching procedures considering the blocking between carts.

Algorithm 1 Two Procedures in FFD

Step 1: Construct two lists that are the candidate bins and bulk units of SKUs

(Binning procedure)

Step 2: If no SKU remains in the bulk units of SKUs, terminate; otherwise, divide the SKUs composed of bulk units into bins per aisle.

Step 3: Sort in ascending order according to the volume of bins for each aisle.

(Batching procedure)

Step 4: Select the shortest route in the route-set to all bins by referring to the location of cells that the bin should visit.

Step 5: If no bin remains in the list of bins, terminate; otherwise, assign the bins in each route to batches considering the batch capacity.

Step 6: Reconstruct the bins in the batch configured at the end of each route to reduce the total number of batches.

Step 7: Sum of the total travel distance of all batches and obtain a solution to the FFD

APPENDIX

A. LINEAR PROGRAMMING RELAXATION

We use the linear programming (LP) relaxation of the BBM-RP model (Section V.A) to derive a lower bound (LB) model by relaxing the integer restrictions.

$$\text{LB}: \min \sum_{r\in R} LT_r \cdot y_r, \tag{20}$$

subject to constraints(14), (15), and(16)

$$x_{sar} \le y_r, \forall s\in S, \forall a\in A, \forall r\in R, \tag{21}$$
$$0 \le x_{sar} \le 1, \forall s\in S, \forall a\in A, \forall r\in R, \tag{22}$$
$$0 \le y_r, \forall r\in R. \tag{23}$$

After LP relaxation, x_{sar} becomes the portion of SKU s's bin at aisle a at route r (Constraint (22)) and y_r becomes the number of SKUs assigned to route r (Constraint (23)). Constraint (21) ensures that if SKU s's bin that should visit aisle a is assigned to route r, there is at least one batch within route r. The LP relaxation of the BBM-RP model by Constraints (22) and (23) provides a weak lower bound.

The valid inequalities based on Constraint (21) enforces y_r to be equal to or greater than maximal x_{sar} for route r and strengthens the lower bound.

B. SHORTEST ROUTE FIRST WITH FIRST-FIT DECREASING

We use Algorithm A1 to quickly construct batches for large-sized problems. We term the heuristic solution the Shortest route first with first-fit decreasing (FFD) [4]. The SKUs delivered in bulk units to the DC need binning in consideration of the quantity (volume) required by the order lists of customers located in each aisle. The bins are distributed in batches to shorten the workers' total travel distance.

Batches are formed depending on the routes available to the workers in compliance with the routing policy and the cart capacity. The FFD consists of a binning and a batching procedure. The binning procedure writes the lists of candidate bins to be covered by each route. The batching procedure constructs the batches using the lists of candidate bins for each route. We assume that all SKUs are split into bins by the number of aisles.

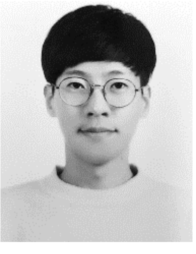
ACKNOWLEDGMENT

An earlier version of this paper titled ''Modelling a batch assorting operation for an autonomous cart in a parallel-aisle order assorting system'' was presented at the 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE) [DOI: 10.1109/COASE.2019.8843189]. *(Taehoon Lee and Jeongman Lee are co-first authors.)*

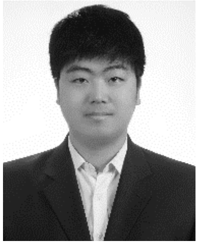
REFERENCES

[1] S. Hong, ''Performance evaluation of two-worker operations in a worker-to-cell order assorting system,'' *J. Manuf. Syst.*, vol. 56, pp. 414–424, Jul. 2020.
[2] M. G. Sternbeck and H. Kuhn, ''An integrative approach to determine store delivery patterns in grocery retailing,'' *Transp. Res. E. Logistics Transp. Rev.*, vol. 70, pp. 205–224, Oct. 2014.
[3] N. Boysen, R. de Koster, and F. Weidinger, ''Warehousing in the e-commerce era: A survey,'' *Eur. J. Oper. Res.*, vol. 277, no. 2, pp. 396–411, Sep. 2019.
[4] J. Lee, Y. Kim, and S. Hong, ''Modelling a batch assorting operation for an autonomous cart in a parallel-aisle order assorting system,'' in *Proc. IEEE 15th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2019, pp. 60–65, doi: 10.1109/COASE.2019.8843189.
[5] J. Gu, M. Goetschalckx, and L. F. McGinnis, ''Research on warehouse design and performance evaluation: A comprehensive review,'' *Eur. J. Oper. Res.*, vol. 203, no. 3, pp. 539–549, Jun. 2010.
[6] K. J. Roodbergen and I. F. A. Vis, ''A survey of literature on automated storage and retrieval systems,'' *Eur. J. Oper. Res.*, vol. 194, no. 2, pp. 343–362, Apr. 2009.
[7] T. van Gils, K. Ramaekers, A. Caris, and R. B. M. de Koster, ''Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review,'' *Eur. J. Oper. Res.*, vol. 267, no. 1, pp. 1–15, May 2018.
[8] R. de Koster, T. Le-Duc, and K. J. Roodbergen, ''Design and control of warehouse order picking: A literature review,'' *Eur. J. Oper. Res.*, vol. 182, no. 2, pp. 481–501, Oct. 2007.
[9] Z. Tan, H. Li, and X. He, ''Optimizing parcel sorting process of vertical sorting system in e-commerce warehouse,'' *Adv. Eng. Informat.*, vol. 48, Apr. 2021, Art. no. 101279.
[10] N. Boysen, D. Briskorn, S. Fedtke, and M. Schmickerath, ''Automated sortation conveyors: A survey from an operational research perspective,'' *Eur. J. Oper. Res.*, vol. 276, no. 3, pp. 796–815, Aug. 2019.
[11] S. Fedtke and N. Boysen, ''Layout planning of sortation conveyors in parcel distribution centers,'' *Transp. Sci.*, vol. 51, no. 1, pp. 3–18, Feb. 2017.
[12] M. E. Johnson and R. D. Meller, ''Performance analysis of split-case sorting systems,'' *Manuf. Service Oper. Manage.*, vol. 4, no. 4, pp. 258–274, Oct. 2002.
[13] D. Agustina, C. K. M. Lee, and R. Piplani, ''Vehicle scheduling and routing at a cross docking center for food supply chains,'' *Int. J. Prod. Econ.*, vol. 152, pp. 29–41, Jun. 2014.
[14] F. Enderer, C. Contardo, and I. Contreras, ''Integrating dock-door assignment and vehicle routing with cross-docking,'' *Comput. Oper. Res.*, vol. 88, pp. 30–43, Dec. 2017.
[15] V. F. Yu, P. Jewpanya, and A. A. N. P. Redi, ''Open vehicle routing problem with cross-docking,'' *Comput. Ind. Eng.*, vol. 94, pp. 6–17, Apr. 2016.
[16] W. Nassief, I. Contreras, and B. Jaumard, ''A comparison of formulations and relaxations for cross-dock door assignment problems,'' *Comput. Oper. Res.*, vol. 94, pp. 76–88, Jun. 2018.
[17] D. Molavi, A. Shahmardan, and M. S. Sajadieh, ''Truck scheduling in a cross docking systems with fixed due dates and shipment sorting,'' *Comput. Ind. Eng.*, vol. 117, pp. 29–40, Mar. 2018.
[18] F. Weidinger, N. Boysen, and M. Schneider, ''Picker routing in the mixed-shelves warehouses of e-commerce retailers,'' *Eur. J. Oper. Res.*, vol. 274, no. 2, pp. 501–515, Apr. 2019.
[19] M. Foumani, A. Moeini, M. Haythorpe, and K. Smith-Miles, ''A cross-entropy method for optimising robotic automated storage and retrieval systems,'' *Int. J. Prod. Res.*, vol. 56, no. 19, pp. 6450–6472, Oct. 2018.
[20] J. Kim and S. Hong, ''A dynamic storage location assignment model for a progressive bypass zone picking system with an S/R crane,'' *J. Oper. Res. Soc.*, pp. 1–12, Mar. 2021, doi: 10.1080/01605682.2021.1892462.
[21] N. Boysen, D. Briskorn, and S. Emde, ''Parts-to-picker based order processing in a rack-moving mobile robots environment,'' *Eur. J. Oper. Res.*, vol. 262, no. 2, pp. 550–562, Oct. 2017.
[22] T. Lamballais, D. Roy, and M. B. M. De Koster, ''Estimating performance in a robotic mobile fulfillment system,'' *Eur. J. Oper. Res.*, vol. 256, no. 3, pp. 976–990, Feb. 2017.
[23] H.-J. Kim, C. Pais, and Z.-J.-M. Shen, ''Item assignment problem in a robotic mobile fulfillment system,'' *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 4, pp. 1854–1867, Oct. 2020.
[24] B. Zou, X. Xu, Y. Gong, and R. De Koster, ''Evaluating battery charging and swapping strategies in a robotic mobile fulfillment system,'' *Eur. J. Oper. Res.*, vol. 267, no. 2, pp. 733–753, Jun. 2018.
[25] A. Bolu and O. Korcak, ''Adaptive task planning for multi-robot smart warehouse,'' *IEEE Access*, vol. 9, pp. 27346–27358, 2021.
[26] D. Roy, S. Nigam, R. de Koster, I. Adan, and J. Resing, ''Robot-storage zone assignment strategies in mobile fulfillment systems,'' *Transp. Res. E, Logistics Transp. Rev.*, vol. 122, pp. 119–142, Feb. 2019.
[27] A. Gharehgozli and N. Zaerpour, ''Robot scheduling for pod retrieval in a robotic mobile fulfillment system,'' *Transp. Res. E, Logistics Transp. Rev.*, vol. 142, Oct. 2020, Art. no. 102087.
[28] E. L. Lawler and D. E. Wood, ''Branch-and-bound methods: A survey,'' *Oper. Res.*, vol. 14, no. 4, pp. 699–719, Jul. 1966.
[29] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance, ''Branch-and-price: Column generation for solving huge integer programs,'' *Oper. Res.*, vol. 46, no. 3, pp. 316–329, May 1998.
[30] M. E. Lübbecke and J. Desrosiers, ''Selected topics in column generation,'' *Oper. Res.*, vol. 53, no. 6, pp. 1007–1023, Dec. 2005.
[31] A. J. R. M. Gademann, J. P. Van Den Berg, and H. H. Van Der Hoff, ''An order batching algorithm for wave picking in a parallel-aisle warehouse,'' *IIE Trans.*, vol. 33, no. 5, pp. 385–398, May 2001.
[32] N. Gademann and S. Velde, ''Order batching to minimize total travel time in a parallel-aisle warehouse,'' *IIE Trans.*, vol. 37, no. 1, pp. 63–75, Jun. 2005.
[33] İ. Muter and T. Öncan, ''An exact solution approach for the order batching problem,'' *IIE Trans.*, vol. 47, no. 7, pp. 728–738, Jul. 2015.
[34] İ. Muter and T. Öncan, ''Order batching and picker scheduling in warehouse order picking,'' *IISE Trans.*, vol. 54, no. 5, pp. 435–447, Jun. 2021.
[35] D. R. Gibson and G. P. Sharp, ''Order batching procedures,'' *Eur. J. Oper. Res.*, vol. 58, no. 1, pp. 57–67, Apr. 1992.
[36] M. B. M. De Koster, E. S. Van der Poort, and M. Wolters, ''Efficient orderbatching methods in warehouses,'' *Int. J. Prod. Res.*, vol. 37, no. 7, pp. 1479–1504, May 1999.
[37] G. Clarke and J. W. Wright, ''Scheduling of vehicles from a central depot to a number of delivery points,'' *Oper. Res.*, vol. 12, no. 4, pp. 568–581, 1964.
[38] S. Hong, A. L. Johnson, and B. A. Peters, ''Large-scale order batching in parallel-aisle picking systems,'' *IIE Trans.*, vol. 44, no. 2, pp. 88–106, Feb. 2012.

[39] S. Hong and Y. Kim, "A route-selecting order batching model with the S-shape routes in a parallel-aisle order picking system," *Eur. J. Oper. Res.*, vol. 257, no. 1, pp. 185–196, Feb. 2017.
[40] C.-M. Hsu, K.-Y. Chen, and M.-C. Chen, "Batching orders in warehouses by minimizing travel distance with genetic algorithms," *Comput. Ind.*, vol. 56, no. 2, pp. 169–178, Feb. 2005.
[41] J. C.-H. Pan, P.-H. Shih, and M.-H. Wu, "Order batching in a pick-and-pass warehousing system with group genetic algorithm," *Omega*, vol. 57, pp. 238–248, Dec. 2015.
[42] M. Matusiak, R. de Koster, L. Kroon, and J. Saarinen, "A fast simulated annealing method for batching precedence-constrained customer orders in a warehouse," *Eur. J. Oper. Res.*, vol. 236, no. 3, pp. 968–977, Aug. 2014.
[43] O. Kulak, Y. Sahin, and M. E. Taner, "Joint order batching and picker routing in single and multiple-cross-aisle warehouses using cluster-based Tabu search algorithms," *Flexible Services Manuf. J.*, vol. 24, no. 1, pp. 52–80, Mar. 2012.
[44] J. Li, R. Huang, and J. B. Dai, "Joint optimisation of order batching and picker routing in the online retailer's warehouse in China," *Int. J. Prod. Res.*, vol. 55, no. 2, pp. 447–461, Jan. 2017.
[45] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to NP-Completeness*. New York, NY, USA: W. H. Freeman, 1979.



TAEHOON LEE received the B.S. degree in industrial engineering from the Kumoh National Institute of Technology and the M.S. degree in industrial engineering from Pusan National University, South Korea, where he is currently pursuing the Ph.D. degree with the Department of Industrial Engineering. His research interest includes optimization and simulation of material handling operations in distribution centers.



JEONGMAN LEE received the B.S. degrees in business administration and industrial engineering and the M.S. degree in industrial engineering from Pusan National University, South Korea, where he is currently pursuing the Ph.D. degree with the Department of Industrial Engineering. His research interest includes analysis and optimization of material handling operations in distribution centers.

SOONDO HONG (Member, IEEE) received the B.S. and M.S. degrees in industrial engineering from POSTECH, South Korea, and the Ph.D. degree from the Department of Industrial and Systems Engineering, Texas A&M University. He worked on scheduling and material handling at semiconductor and display companies. He is an Associate Professor with the Department of Industrial Engineering, Pusan National University, South Korea. His research interests include logistics operations and material handling in manufacturing, warehousing, display, semiconductor, and container terminal industries.