Computers & Industrial Engineering 129 (2019) 210–219

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

New model of the storage location assignment problem considering demand correlation pattern

Ren-Qian Zhang[a], Meng Wang[a], Xing Pan[b,*]

[a] School of Economics and Management, Beihang University, Beijing 100191, China
[b] School of Reliability and System Engineering, Beihang University, Beijing 100191, China

ARTICLE INFO

Keywords:
Order-picking
Storage location assignment
Demand correlation
Logistics

ABSTRACT

Order-picking is the most time- and labor-consuming operation in a warehouse and significantly influences supply chain efficiency. One of the basic methods for improving order-picking efficiency involves assigning storage locations to appropriate items, i.e., the storage location assignment problem (SLAP). In existing studies, most storage assignment methods only consider the properties of individual item rather than the item groups that are usually collectively required. This paper introduces the concept of the demand correlation pattern (DCP) to describe the correlation among items, based on which a new model is constructed to address the SLAP. The model is subsequently reduced using the S-shape routing strategy, and a method for determining DCPs from historical data is proposed. To solve the model, a heuristic and a simulated annealing method are developed. The proposed methods are examined and compared extant methods using both real data collected from an online retailer and numerical instances that are randomly generated. The computational results are discussed.

1. Introduction

A warehouse is an intermediary facility between suppliers and customers that has an important role in daily supply chain operations. Warehouse activities can be decomposed into receiving, storing, order-picking, sorting, and shipping, among which order-picking is the most time- and labor-consuming operation. According to Richards (2014), order-picking accounts for approximately 35% of warehouse operating costs. For companies faced with cost reduction, improving order-picking efficiency may be one of the most effective options.

Order-picking is the process of retrieving items from storage locations to fulfill customers' orders. Two types of order-picking systems usually exist in practice: *parts-to-picker* systems and *picker-to-parts* systems. The parts-to-picker system employs automated storage/retrieval (AS/R) machines or automated guided vehicles (AGVs) to carry the required items to pickers, which has been intensively investigated in the literature (Boysen, De Koster, & Weidinger, in press; Boze & Aldarondo, 2018; Roodbergen & Vis, 2009). In a picker-to-parts system, pickers travel along aisles to retrieve all required items from storage locations. Due to the greater flexibility of humans in adapting to real-time changes and less investment, this manual system forms a large proportion of all order-picking systems worldwide. Marchet, Melacini, and Perotti (2015) investigate 40 companies and find that half of them are equipped with the picker-to-parts order-picking systems. Napolitano (2012) shows that in 2011–2012, about 70–80% materials handling systems have conventional picking system. Dallari, Marchet, and Melacini (2009) investigate 68 warehouse facilities of big companies and find that the picker-to-parts system is very popular in the situations where the number of items and picking volume (order line per day) are within 1000 and 10,000. These results are consistent with the case of JD.com, Inc. (JD), which is one of the largest e-commerce companies in China. Although fully automated warehouses (such as the JD Asia No.1 Warehouse) are possible, we discovered that a large proportion of warehouses remain manually operated. We consider the picker-to-parts system in this paper.

The order-picking efficiency can be improved by a number of approaches, including (1) assigning items to appropriate storage locations (*storage strategy*); (2) determining appropriate route of the picking tour (*routing*), and (3) picking orders in batches (*batching*) (Boysen et al., in press; Gils, Ramaekers, Caris, & De Koster, 2018). Among the three approaches, the storage strategy is the basis of routing and batching since the latter approaches have to be performed with the aid of storage information. This research focuses on improving the order-picking efficiency by developing efficient storage location assignment, whereas the picker routing problem is solved via a heuristic and batching is not considered.

The storage location assignment problem (SLAP) involves determining the allocation of items to storage locations to maximize the

* Corresponding author.
E-mail address: panxing@buaa.edu.cn (X. Pan).

https://doi.org/10.1016/j.cie.2019.01.027
Received 22 May 2018; Received in revised form 23 December 2018; Accepted 13 January 2019
Available online 18 January 2019
0360-8352/

order-picking efficiency. Classic storage strategies usually assign items according to their individual properties, such as popularity (Frazelle, 2002), picking density (Petersen, Siu, & Heiser, 2005), turnover (Hausman, Schwarz, & Graves, 1976), volume (Petersen & Schmenner, 1999) or cube-per-order index (COI, Heskett, 1963). Class-based storage strategy that divides items into several classes and assigns each class to a dedicated storage area is one of the most commonly adopted storage strategies in practice (Dijkstra & Roodbergen, 2017; Muppani & Adil, 2008). Some researchers also make assignment decisions by considering the item correlation, where items are thought to be *correlated* if they appear in the same order at high frequency. Correlation-based storage strategies (also referred to as *family grouping* by De Koster, Le-Duc, & Roodbergen (2007)) place correlated items as near to each other as possible to ensure that they can be picked together without traveling a long distance; as a result, the picking effort is reduced. The results of Ruben and Jacobs (1999), Bindi, Manzini, Pareschi, and Regattieri (2009) and Glock and Grosse (2012) indicate that the correlation-based strategy performs better than the class-based strategy in their picking circumstances.

To place correlated items close to each other, clustering-assigning method is commonly used, in which highly correlated items are clustered into groups and groups are then assigned to storage locations (Chiang, Lin, & Chen, 2014; Chuang, Lee, & Lai, 2012; Liu, 2004; Xiao & Zheng, 2010). The majority of studies use *pairwise correlation* to develop their correlation-based storage strategies by referring to the statistical correlation between any two items, which is an indirect description of the correlation among three or more items.

Instead of using pairwise correlation, we introduce the concept of the demand correlation pattern (DCP) to describe item correlation. Based on the DCP, a mathematical model for SLAP is constructed. Due to the complexity of the problem, we solve it heuristically. Two heuristics are developed and evaluated using real data from an online retailer and numerical instances that are randomly generated. The purposes of this paper are described as follows: (1) present a new measurement for item correlation, based on which a new model for the SLAP is built, (2) develop two heuristics to solve the problem, and (3) investigate the proposed methods against extant storage strategies in different environments.

The remainder of the paper is organized as follows. In the next section, a detailed literature review is provided and related studies are discussed. In Section 3, the SLAP is formulated as a new model based on the DCP. To solve the model, a heuristic and a simulated annealing method are developed in Section 4. Experiments are conducted to examine the proposed methods, and the computational results are discussed in Section 5. Section 6 states the conclusions.

2. Literature review

In this section, we discuss the related studies in two aspects: storage strategies, especially correlation-based strategies, and simulated annealing methods used in order-picking research.

2.1. Storage strategies

A storage strategy is a set of rules that assign items to storage locations to improve the order-picking efficiency. Basic strategies include the *random strategy*, *dedicated strategy*, and *class-based strategy*. The random strategy randomly assigns arriving pallets to available storage locations to enable the storage space to be shared, which causes high storage utilization and low order-picking efficiency (De Koster et al., 2007). The dedicated strategy assigns each item to a dedicated storage location according to its individual properties, such as popularity, turnover, volume or COI (Petersen et al., 2005). One advantage of the dedicated strategy is that pickers become familiar with the locations of items and as a result, the picking efficiency is improved at the price of low storage utilization (De Koster et al., 2007). The class-based strategy divides items into a certain number of classes and subsequently assigns each class to a dedicated area in the warehouse. Within each area, the storage locations are randomly assigned. The class-based strategy is a tradeoff between the storage-space cost and the order-picking cost (Muppani & Adil, 2008). The effectiveness of the three basic strategies was examined in Hausman et al. (1976), Petersen and Aase (2004), Muppani and Adil (2008), Chan and Chan (2011) and Dijkstra and Roodbergen (2017).

Classic storage strategies perform well in the case in which only one item is picked in each tour (Malmborg & Bhaskaran, 1990). In reality, however, multiple items within an order are often collectively picked and the efficiency can be improved by identifying the correlated items and placing them close to each other. Frazelle and Sharp (1989) extract the pairwise statistical correlation from historical orders and formulate the SLAP as an integer programming problem, which is shown to be NP-hard and solved using a two-stage heuristic. Lee (1992) studies the man-on-board AS/R system, in which a clustering technology that employs the Hungarian method is proposed and items are located following the space-filling curves. Kim (1993) develops a heuristic for the SLAP model considering the inventory-related cost and material-handling cost. Brynzér and Johansson (1996) propose a strategy that emanates from the product structure, which reduces the amount of picking information to the picker by 75%. Liu (1999) and Liu (2004) propose the order-item-quantity rule to measure the similarity between two items and locate items to storage locations along the direction of routing strategy, which is a commonly employed positioning method. Jane and Laih (2005) investigate the synchronized zone order-picking system and develop a heuristic to assign correlated items to different zones so that the picking operation can simultaneously proceed. Hua and Zhou (2008) explore SLAP in a circuit board assembly kitting area, where components have to be effectively collected. The authors present a new clustering method and position item groups following specific filling-curves. Bindi et al. (2009) evaluate the performance of correlation-based storage strategies with different similarity measurements, group assignment rules, routing policies and warehouse shapes. Xiao and Zheng (2010) investigate a production warehouse in which parts, as determined by the bill of materials (BOM), must be picked from storage locations. A multi-stage heuristic, in which the clustering method is modified from Lee (1992), is developed. Chuang et al. (2012) improve family grouping by introducing the idea of between-item association and construct mixed integer programming formulations for both item clustering and group locating problems, which are linearized and solved via Lingo.

Instead of the clustering-assigning method, several different approaches have been proposed in recent years. Pang and Chan (2017) consider the put-away cost in their model and propose a data mining-based algorithm, that extracts and analyses the association relationships between different products in customer orders. Chiang, Lin, and Chen (2011) propose the concept of the association index (AIX), which evaluates the fitness between items and available storage locations, to determine the optimal storage assignment for newly delivered items to be put away. The study is subsequently extended to solve the general SLAP in Chiang et al. (2014), in which two heuristics are proposed. Glock and Grosse (2012) investigate the U-shaped order-picking system, in which the item demand and correlation are considered, and propose a storage assignment method that is based on pair correlation. Our paper considers both demand and correlation, but employs different correlation measurement and methods. Wutthisirisart, Noble, and Chang (2015) adopt the concept of linear placement as an approximate description for a real warehouse layout to create a minimum delay algorithm (MDA) that considers both item relationships and order characteristics. Li, Moghaddam, and Nof (2015) propose a mixed integer programming model for the dynamic storage location assignment problem which is solved by a greedy genetic algorithm. The previously mentioned correlation-based strategies are summarized in Table 1. In the column of similarity measurement, $\mathbf{z}_i = (z_{i1}, ..., z_{iK})$ indicates the

Table 1
Summary of correlation-based storage strategies.

Paper	Order-picking system	Routing strategy	Similarity measurement	Method
Lee (1992)	Man-on-board AS/R system	–	$\frac{\sum_k (f_k \sqrt{z_{ik} z_{jk}})}{\sum_k \sqrt{f_k (z_{ik} + z_{jk})}}$ [a]	Hungarian method based clustering algorithm; Filling curve based assignment.
Kim (1993)	Miniload AS/R system	–	–	Clustering heuristic minimizing the model's objective value
Liu (2004, 1999)	Manual system with gravity-flow racks	Z type route	$\frac{1}{K} \sum_k \frac{\min\{z_{ik}, z_{jk}\}}{\max\{z_{ik}, z_{jk}\}}$	Ranking, clustering and interchanging heuristic; Primal-dual algorithm
Jane and Laih (2005)	Synchronized zone order-picking system	–	$\sum_k z_{ik} z_{jk}$	Natural clustering algorithm
Xiao and Zheng (2010)	Picker-to-parts system	S-shape, Largest gap	$\frac{\sum_k f_k (z_{ik} \wedge z_{jk})}{\sum_k f_k (z_{ik} \vee z_{jk})}$	Hungarian method based clustering algorithm; Assignment along the picking route
Chuang et al. (2012)	Manual system with single aisle	Z type route	$1 - \frac{n_{ij}}{n_i + n_j}$	Linearized and solved by Lingo
Pang and Chan (2017)	Picker-to-parts system	Insertion heuristic	$\frac{n_{ij}}{n_i}, \frac{n_{ij}}{n_j}$	Solved by relaxing the model
Bindi et al. (2009)	Picker-to-parts system	S-shape, Return	$\frac{n_{ij} \min\{T_i, T_j\}}{[n_{ij} + 0.25(n_i + n_j)] \max\{T_i, T_j\}}$ [b]	Hierarchical clustering algorithms; Positioning rules: Zig-zag, Stripes
Chiang et al. (2014)	Picker-to-parts system	S-shape	WSC [c]	Modified class-based heuristic; Association seed based heuristic; Genetic algorithm
Wutthisirisart et al. (2015)	Picker-to-parts system	S-shape	–	Minimum delay algorithm
Glock and Grosse (2012)	U-shaped order-picking system	Sweep algorithm	I_1, I_2 [d]	An iterated assigning heuristic based on I_1 and I_2.
This paper	Picker-to-parts system	S-shape	Demand correlation pattern that is extracted from historical data	Minimum increment heuristic and simulated annealing method

[a] f_k refers to the frequency of order k.
[b] T_i=Total movement/ Average stock quantity; refer to Bindi et al. (2009) for details.
[c] WSC= n_{ij}, 0, $-n_{ij}$ with respect to $n_{ij}/(n_i n_j) >, =, < 1$, respectively.

appearance of item i in the order set, where $z_{ik} = 1$ if item i is contained by order $k = 1, ..., K$; 0 otherwise. Let $\wedge$ and $\vee$ represent the operators, AND and OR. In addition, $n_i = |\mathbf{z}_i|$ represents the number of orders that contain item i and $n_{ij} = |\mathbf{z}_i \wedge \mathbf{z}_j| = \sum_k z_{ik} \wedge z_{jk}$ represents the number of orders that simultaneously contain i and j.

Since picker-to-parts systems are primarily operated by humans, it is reasonable and practical to consider human factors in the SLAP. Petersen et al. (2005) propose several storage strategies that put high demand items at the height between picker's waist and shoulder (*golden zone*) for easier access, and their simulation results show that the golden zone storage strategies generate significant savings in terms of the order fulfillment time. Grosse, Glock, and Jaber (2013) examine the effect of worker learning and forgetting on storage assignment decisions, and their results support managers in terms of when to change or keep a storage assignment if the learning and forgetting occur. Battini, Glock, Grosse, Persona, and Sgarbossa (2016) develop a bi-objective method to the SLAP model that considers both picking time and human energy expenditure, which can cause risky environments for workers to develop musculoskeletal disorders. Another bi-objective model is constructed by Larco, De Koster, Roodbergen, and Dul (2017), who make a tradeoff between warehouse efficiency and worker discomfort via storage assignment decisions. Otto, Boysen, Scholl, and Walter (2017) define a combined ergonomic storage assignment and zoning model to minimize the maximum ergonomic burden among all workers in the picking area. Three solution methods are developed, and their results show that much higher ergonomic risks exist if the ergonomic aspects are neglected. For studies that consider human factors in other order-picking problems, please see, Elbert, Franzke, Glock, and Grosse (2017), Glock, Grosse, Elbert, and Franzke (2017), Grosse, Glock, and Neumann (2016) and Grosse, Glock, Jaber, and Neumann (2015).

2.2. Simulated annealing method

Due to the NP-hardness of many order-picking problems, such as the SLAP or order batching problem (OBP), they are often solved by heuristics or meta-heuristics. Heuristics for the SLAP, including clustering methods and assigning rules, are summarized in Table 1.

Simulated annealing (SA) has been applied to effectively solving many combinatorial problems. In the literature, several SA approaches have been proposed for order-picking problems. Lai, Xue, and Zhang (2002) propose SA methods to assign items to appropriate cells in a paper reel warehouse. Muppani and Adil (2008) consider the holding cost in a class-based storage policy and propose an SA approach to determine the best members for each class by randomly moving items from one class to another. Bartholdi and Gue (2000) use an SA approach to optimize the layout of trailers in a crossdocking terminal to reduce the labor cost and congestion. Matusiak, De Koster, Kroon, and Saarinen (2014) present a fast SA method for the OBP, in which the precedence of orders is considered, that obtains better solutions by randomly forming new batches from the old batches. Another SA approach for the OBP is discussed in Grosse, Glock, and Ballester-Ripoll (2014).

Instead of the pairwise correlation measurement that is commonly employed in the literature, this paper introduces a new correlation measurement, based on which a new mathematical model is constructed. We develop a heuristic and an SA method to solve the problem.

3. Model

In this section, the concept of DCP is introduced to describe how items are correlated, and then a mathematical model for the SLAP is constructed (Section 3.1). Due to the high complexity in determining the picking route for each order, we reduce the model with the S-shape routing heuristic (Section 3.2). We demonstrate how to identify the DCPs of items from historical data in Section 3.3.

3.1. Problem formulation

We examine the SLAP in a low-level picker-to-parts warehouse, which is commonly adopted in practice (De Koster et al., 2007). The warehouse configuration remains the same as the configuration in Xiao

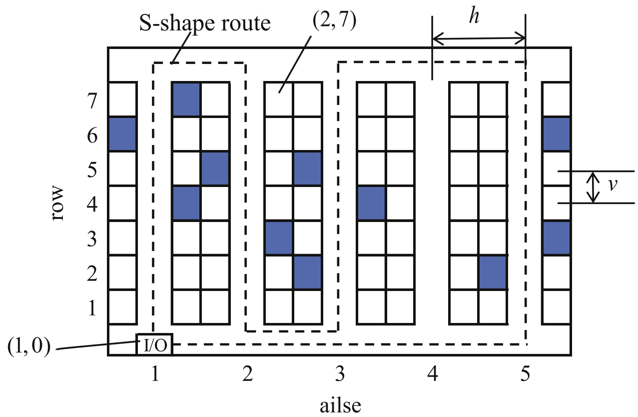

Fig. 1. Warehouse layout and S-shape route.

and Zheng (2010), where multiple aisles exist and storage locations are distributed on both sides of each aisle. The layout is shown in Fig. 1. Assume that N types of items are to be assigned to L storage locations and the objective is to determine an assignment (solution) that minimizes the total traveling distance.

We assume that one type of item occupies exactly one storage location and that one storage location holds only one type of item. In other words, stock mixing or splitting is not allowed; this assumption is also applied in Xiao and Zheng (2010) and Chiang et al. (2014). The space of any storage location is sufficiently large for storing all quantities of items of the same type. Since the warehouse is two-dimensional and different types of items are placed in different locations, travel distance will be incurred by moving from one item to the next item. We also assume that the replenishment cost is disregarded, since the cost is minimal compared with the cost of order-picking due to bulk replenishment (Ruben & Jacobs, 1999).

A customer who buys one item may tend to simultaneously buy other items. The demand correlation among items is common in practice. For example, the components that are required by one product or materials that belong to the same BOMs are always simultaneously picked. The demand dependency often results in the correlation among three or more items, which implies that the pairwise correlation measure may be not the best choice. We introduce the concept of DCP to describe the previously mentioned item correlation. The DCP of item i is defined as the set of items that are simultaneously ordered (with a certain probability) due to the requirement of i. Item i is highly correlated with other items in its DCP. Evidently, for item i, more than one DCPs may exist, the set of which is denoted by $\mathcal{M}_i$. For example, $\mathcal{M}_1 = \{\varnothing, \{2\}, \{2, 5\}, \{3, 4\}\}$ represents all DCPs of item 1. If item 1 is ordered, then the DCP of {2}, {2,5} or {3,4} can also be ordered, and the probabilities are determined from historical data, as illustrated in Section 3.3. Specifically, DCP = $\varnothing$ indicates that item 1 is independently ordered. The associated notations are as follows:

$\mathcal{N} = \{1, ..., N\}$: set of items, indexed by $i, j \in \mathcal{N}$;
$\mathcal{L} = \{1, ..., L\}$: set of storage locations, indexed by $k, l \in \mathcal{L}$;
$\mathcal{O} = \{1, ..., O\}$: set of historical orders, indexed by $o \in \mathcal{O}$;
$\mathcal{A} = \{1, ..., A\}$: set of aisles, indexed by $a \in \mathcal{A}$;
$\mathcal{R} = \{1, ..., R\}$: set of rows, indexed by $r \in \mathcal{R}$;
$\mathcal{M}_i$: set of DCPs of item $i \in \mathcal{N}$, indexed by $m \in \mathcal{M}_i$;
μ: the I/O point. Let $\mathcal{L}^+ = \mathcal{L} \cup \{\mu\}$;
D_i: demand of item i;
$a_k \in \mathcal{A}$: aisle number of storage location k;
$r_k \in \mathcal{R}$: row number of storage location k;
d_{kl}: distance between storage location k and l;
v: distance between two adjacent storage locations;
h: distance between two adjacent aisles;
p_m: probability that DCP m is ordered;
$e_{mj} = 1$: if item j is included in DCP m ($j \in m$); 0 otherwise.
$|\cdot|$: the cardinality of the given set.

Decision variables:

$x_{ik} = 1$: if item i is assigned to location k; 0, otherwise;
$y_{mkl} = 1$: if storage location l is visited immediately after k in the route for picking the items in DCP m; 0, otherwise;

To minimize the total travel distance (TTD) for picking all DCPs, the demand correlation pattern based model (DCPM) of the SLAP is constructed.

$$\min TTD = \sum_{i\in\mathcal{N}} \sum_{m\in\mathcal{M}_i} \sum_{k\in\mathcal{L}^+} \sum_{l\in\mathcal{L}^+} D_i p_m d_{kl} y_{mkl}, \tag{1}$$

s.t.

$$\sum_{i\in\mathcal{N}} x_{ik} = 1, \quad k \in \mathcal{L}, \tag{2}$$

$$\sum_{k\in\mathcal{L}} x_{ik} = 1, \quad i \in \mathcal{N}, \tag{3}$$

$$\sum_{l\in\mathcal{L}} y_{m\mu l} = 1, \quad m \in \mathcal{M}_i, i \in \mathcal{N}, \tag{4}$$

$$\sum_{k\in\mathcal{L}} y_{mk\mu} = 1, \quad m \in \mathcal{M}_i, i \in \mathcal{N}, \tag{5}$$

$$\sum_{k\in\mathcal{L}^+} y_{mkl} \leqslant 1, \quad l \in \mathcal{L}, m \in \mathcal{M}_i, i \in \mathcal{N}, \tag{6}$$

$$\sum_{l\in\mathcal{L}^+} y_{mkl} \leqslant 1, \quad k \in \mathcal{L}, m \in \mathcal{M}_i, i \in \mathcal{N}, \tag{7}$$

$$\sum_{k_1\in\mathcal{L}^+} y_{m,k_1,l} = \sum_{k_2\in\mathcal{L}^+} y_{m,l,k_2}, \quad l \in \mathcal{L}, m \in \mathcal{M}_i, i \in \mathcal{N}, \tag{8}$$

$$\sum_{k\in\mathcal{L}^+} \sum_{l\in\mathcal{L}^+} y_{mkl} = \sum_{j\in\mathcal{N}} e_{mj} + 1, \quad m \in \mathcal{M}_i, i \in \mathcal{N}, \tag{9}$$

$$u_k - u_l + \sum_{j\in\mathcal{N}} e_{mj} y_{mkl} \leqslant \sum_{j\in\mathcal{N}} e_{mj} - 1, \quad k, l \in \mathcal{L}^+, k \neq l, \tag{10}$$

$$y_{mkl} \leqslant \sum_{j_1\in\mathcal{N}} e_{m,j_1} x_{j_1,k} \sum_{j_2\in\mathcal{N}} e_{m,j_2} x_{j_2,l}, \quad k, l \in \mathcal{L}, m \in \mathcal{M}_i, i \in \mathcal{N}, \tag{11}$$

$$x_{ik} \in \{0, 1\}, \quad i \in \mathcal{N}, k \in \mathcal{L}, \tag{12}$$

$$y_{mkl} \in \{0, 1\}, \quad m \in \mathcal{M}_i, i \in \mathcal{N}, k, l \in \mathcal{L}^+, \tag{13}$$

where the distance between any two storage locations is calculated as follows:

$$d_{kl} = \begin{cases} |r_l - r_k|\, v, & \text{if } a_l = a_k \\ |a_l - a_k|\, h + \min\{2R - (r_l + r_k) + 2, (r_l + r_k)\}v, & \text{if } a_l \neq a_k \end{cases}$$

Constraints (2) and (3) indicate the assumptions that one storage location only holds one item type and one item type must be assigned to one storage location, respectively. Constraints (4)–(10) guarantee the integrity of the route for picking DCP m, which starts from the I/O point, traverses all storage locations that store the items in DCP m and ends at the I/O point. Specifically, constraints (4) and (5) indicate that the picker must start and finish at the I/O point. Constraints (6) and (7) indicate that at any visited location, with the exception of the I/O point, only an antecedent location and a succedent location exist. Constraints (8) ensure that the input degree must be equal to the output degree for each storage location. Constraints (9) indicates that the number of visited storage locations (with the exception of the I/O point) is equal to the number of items in DCP m. Constraints (10) derived from the traveling salesman problem (TSP) eliminates sub-cycles where u_k and u_l denote variables of arbitrary values. For constraints (10), the feasible tour must be a cycle that starts from the I/O point, ends at the I/O point and connects all visited locations. Constraints (11) ensure that all visited storage locations must correspondingly store the requested items in DCP m.

3.2. Routing strategy

The complexity of DCPM is high since for any assignment $\{x_{ik}: i \in \mathcal{N}, k \in \mathcal{L}\}$, there are a total of $\sum_i \mathcal{M}_i$ sub-problems to be solved to determine the route $\{y_{mkl}: k, l \in \mathcal{L}\}$ for picking DCP $m \in \mathcal{M}_i$. The sub-problem is a special case of TSP and can be optimized by the algorithms proposed by Ratliff and Rosenthal (1983) and Scholz, Henn, Stuhlmann, and Wäscher (2016). Although the computational effort of the algorithms are polynomials, they are still too "expensive" for the DCPM, and the optimal route is typically "illogical" to pickers (Elbert et al., 2017; Glock et al., 2017; Hall, 1993). Therefore, the sub-problem is usually solved via heuristics. The commonly used routing heuristics include the *S-shape*, *return*, *mid-point*, and *largest gap* heuristics (De Koster et al., 2007). Petersen and Schmenner (1999) and Dijkstra and Roodbergen (2017) examine the joint effects of routing strategies and storage location assignments; the conclusions indicate that the final storage location assignment adapts to the employed routing strategies. The performance of the final storage location assignment, i.e., *TTD*, does not significantly vary when different routing heuristics are selected.

In this paper, the S-shape heuristic is employed for the subproblem in which aisles that contain at least one requested item are entirely traversed (with the exception of the last visited aisle, where the picker needs to turn back if the number of visited aisles is odd) and aisles without requested items are not entered. An important reason for choosing the S-shape is that, for this routing strategy, each picker possesses only one direction, which will reduce the aisle congestion (disregarded in the study) when the pick density is sufficiently high (De Koster et al., 2007). The S-shape routing strategy is illustrated in Fig. 1.

With the S-shape routing strategy, the DCPM is reduced. With respect to a certain storage location assignment $\{x_{ik}, i \in \mathcal{N}, k \in \mathcal{L}\}$, the visited storage locations for picking DCP m are determined as $\mathcal{L}_m = \{k \in \mathcal{L}: x_{jk} = 1, j \in m\}$ and subsequently the visited aisles are $\mathcal{A}_m = \{a_k \in \mathcal{A}: k \in \mathcal{L}_m\}$. The last visited aisle is $\bar{a}_m = \max\{a_k \in \mathcal{A}_m\}$ and the row of the last visited storage location in aisle $\bar{a}_m$ is $\bar{r}_m = \max\{r_k \in \mathcal{R}: a_k = \bar{a}_m, k \in \mathcal{L}_m\}$. Consequently the travel distance for picking m is recast as:

$$\sum_{k \in \mathcal{L}^+} \sum_{k \in \mathcal{L}^+} d_{kl} y_{mkl} = 2(\bar{a}_m - 1)h + |\mathcal{A}_m|(R+1)v + (2\bar{r}_m - R - 1)^{|\mathcal{A}_m| \bmod 2} v.$$

As a result, we obtain the reduced demand pattern-based model (RDCPM) as follows.

$$\min\ TTD = \sum_{i \in N} \sum_{m \in \mathcal{M}_i} D_i p_m [2(\bar{a}_m - 1)h + |\mathcal{A}_m|(R+1)v + (2\bar{r}_m - R - 1)^{|\mathcal{A}_m| \bmod 2} v] \tag{14}$$

s.t.

(2),(3),(12).

The complexity of the RDCPM is significantly reduced compared with that of the DCPM. The development of an effective algorithm once the DCPs and demands of items are determined is possible.

3.3. Determining DCPs

With respect to item i, numerous associated DCPs may exist and the size of $\mathcal{M}_i$ is extremely high, since (1) a few DCPs only occur with small probabilities and (2) a few DCPs may be the supersets of the other DCPs. For example, in the DCP = {2, 3} of item 1, the item pairs of 1 and 2 or 1 and 3 are correlated, and therefore, both {2} and {3} are also DCPs of item 1. To overcome the weaknesses, only the DCP that satisfies the following conditions is considered: (1) it frequently occurs; and (2) it does not have a superset. The available DCPs can be determined by applying maximal frequent itemset mining algorithms, such as FP-max (Grahne & Zhu, 2003), to historical orders.

Algorithm 1 enables us to determine all available DCP sets $\mathcal{M}_i$, $i = 1, \ldots, N$ from historical orders O, where β_i represents the support count of item i, i.e., the total number of orders that contain i. With the minimum support count $\beta_{\min}$ as the threshold of frequency, all maximal frequent itemsets in O are mined out by FP-max and added to DCP sets based on the rule of maximizing the association confidence (steps 4 and 5 in Algorithm 1).

Algorithm 1. Method for determining available DCP sets

1: For item $i \in \mathcal{N}$, calculate its support count β_i and set $\mathcal{M}_i = \varnothing$
2: Run FP-max on O to determine all maximal frequent itemsets, denoted by $\mathcal{M}$. Each element $m \in \mathcal{M}$ satisfies $\beta_m \geqslant \beta_{\min}$
3: **for** $m \in \mathcal{M}$ **do**
4: Get $i' = \text{argmax}_{i \in m}\left\{\frac{\beta_m}{\beta_i}\right\}$
5: Set $p_m = \frac{\beta_m}{\beta_{i'}}$ and $\mathcal{M}_{i'} := \mathcal{M}_{i'} \cup \{m\}$
6: **end for**

To demonstrate the implementation of Algorithm 1, an example for which the historical data set O is displayed in Table 2 is considered. If we set $\beta_{\min} = 5$ and apply FP-max to O, a maximal frequent itemset is obtained: $m = \{2, 5, 7\}$ with $\beta_m = 5$. Additionally, the support count of the items in m, β_i ($i = 2, 5, 7$) are calculated as 6, 7, 7, respectively. Given that $i' = \text{argmax}_{i=2,5,7}\left\{\frac{\beta_m}{\beta_i}\right\} = 2$, we set $\mathcal{M}_2 := \mathcal{M}_2 \cup \{m\}$ and $p_m = 0.83$. The demand, D_i ($i = 1, \ldots, N$) is estimated by the frequency of i in O; for example, the frequency of item 2 is 8, and thus $D_2 = 8$.

4. Solution

The non-linearity and non-convexity of the proposed model exclude traditional integer programming techniques. Therefore, we will heuristically solve the model. This section presents two methods: the minimum increment heuristic (MIH) and simulated annealing (SA) method.

4.1. Minimum Increment Heuristic (MIH)

MIH assumes that the picking route always starts from the I/O point and traverses all possible aisles until the location of the last item is reached. We refer to the specific picking route as the *all-traverse route*, which ensures that the travel distance is only dependent on the location of the last picked item in the order. Starting with all empty locations and total travel distance of zero, MIH assigns each item to the most

Table 2
An example of historical data O.

Order	Items
o_1	1, 2, 3, 5, 5, 7
o_2	3, 4, 7
o_3	2, 4, 5, 6, 7, 9
o_4	2, 2, 10
o_5	2, 5, 6, 7, 8
o_6	2, 3, 7
o_7	2, 3, 4, 5, 5, 7
o_8	1, 3
o_9	5
o_{10}	1, 2, 5, 7, 8, 10

remote location and selects the one with a minimum increment of travel distance. Consider that the item, say i', with a minimum number of involved orders will be assigned. Since orders that contain the assigned item i' must be completed by reaching the most remote location, they are removed from the order set and are not considered in the next iteration. The process repeats until all items are assigned. MIH is described in Algorithm 2, where $\mathcal{N}'$ represents the set of unassigned items and $\mathcal{L}'$ the set of empty locations. Note that MIH takes advantage of the item correlation by removing orders that contain the assigned items (step 7 in Algorithm 2).

Algorithm 2. MIH

1: Set $\mathcal{N}' := \mathcal{N}$ and $\mathcal{L}' := \mathcal{L}$
2: Calculate the all-traverse route distance d_k' for $k \in \mathcal{L}'$
3: **repeat**
4: From $\mathcal{O}$, calculate the number of involved orders β_i for $i \in \mathcal{N}'$
5: Assign item $i' = \mathrm{argmin}_{i \in \mathcal{N}'} \beta_i$ to location $k' = \mathrm{argmax}_{k \in \mathcal{L}'} d_k'$
6: Set $\mathcal{N}' := \mathcal{N}' - \{i'\}$ and $\mathcal{L}' := \mathcal{L}' - \{k'\}$
7: Remove all orders that contain i' from $\mathcal{O}$
8: **until** $\mathcal{N}' = \varnothing$

4.2. Simulated Annealing (SA)

SA has been applied to solving many combinatorial optimization problems. We observe that feasible solutions can be easily obtained by swapping the storage locations of any two items and that the local search based meta-heuristic can be more flexible and easily to be implemented. This makes SA with a good balance between exploration and exploitation suitable to solve the model.

We construct an initial assignment by assigning items with high picking frequency to aisles near the I/O point as much as possible. The initial solution is refined through the SA procedure that finds neighbors from the current solution and evaluates each neighbor using objective (14). Better neighbors with smaller objective values will be accepted, whereas worse neighbors will be accepted with the probability of $\exp\left(-\frac{\Delta}{T}\right)$, where Δ is the gap of objective values between the neighbor and the current solution, and T the current temperature. The inner loop with the temperature T has *SAIter* iterations, and subsequently the temperature will be cooled by $T := \lambda T$, where $\lambda \in (0, 1)$ is the cooling rate. The commonly used *swapping operator* is employed in this study to obtain the neighbors from the current solution, which randomly selects two items and exchanges their storage locations. SA is described in Algorithm 3, where s_0 and s^* are the initial solution and the best solution, respectively. T_0 and $T_{\min}$ are the initial temperature and terminal temperature, respectively.

A suitable parameter setting typically indicates a good balance between the solution quality and the computational effort. We use a number of experiments and set $T_0 = 800$, $T_{\min} = 1$ and $\lambda = 0.95$. Regarding the inner loop iteration (steps 4–10 in Algorithm 3), we set $SAIter = 2 \times N$ as the neighborhood space grows with an increase in the instances scale. SA terminates when the current temperature is less than $T_{\min}$.

Algorithm 3. SA

1: Set $s := s_0$, $s^* := s_0$ and $T := T_0$;
2: **repeat**
3: Set $u := 0$
4: **repeat**
5: Obtain a neighbor s' by swapping operator
6: Calculate $\Delta = TTD(s') - TTD(s)$
7: If $\Delta \leqslant 0$, set $s := s'$; otherwise, set $s := s'$ with the probability of $\exp\left(-\frac{\Delta}{T}\right)$
8: If $TTD(s) - TTD(s^*) < 0$, set $s^* := s$
9: Set $u := u + 1$
10: **until** $u > SAIter$
11: $T := \lambda T$
12: **until** $T > T_{\min}$

5. Computational study

This section presents several computational experiments to evaluate the proposed methods using both real data collected from an online retailer and numerical data that are randomly generated. The performances of the proposed methods are compared with those of extant methods from the literature. All implemented algorithms are encoded in Java and run on a Windows 10 platform with Intel i7-3770 CPU and 4.0 G Rom. Section 5.1 describes the experiment setting. Section 5.2 and 5.3 present the computational results for the real data and numerical instances, respectively.

5.1. Experiment setting

First, we shall discuss how to assess and compare the performance of the assignment methods and how to determine the minimum support count $\beta_{\min}$. Moreover, the methods used for comparison are described.

Performance evaluation. We compare our methods with existing SLAP methods from the literature. To assess the solution, the training-testing evaluation mechanism is employed, in which the order set in each problem is split into two parts, namely the training part and the testing part. The training part, including 70% of all orders, is utilized as historical data and input into each method to produce a solution, whereas the testing part, including the remaining 30% of orders, is used as newly arrived orders for testing the solution generated by each method. The testing result (*TR*) in terms of the total travel distance for picking all testing orders is the output. The training-testing mechanism is described in Fig. 2. We use the dedicated storage strategy (DSS) as the benchmark, which assigns items with the highest picking frequency to locations nearest to the I/O point. The improvement (*Impr*) over DSS is finally output as the performance index. *Impr* is calculated as:

$$Impr = \frac{TR(\mathrm{DSS}) - TR(\mathrm{method})}{TR(\mathrm{DSS})} \times 100\%, \tag{15}$$

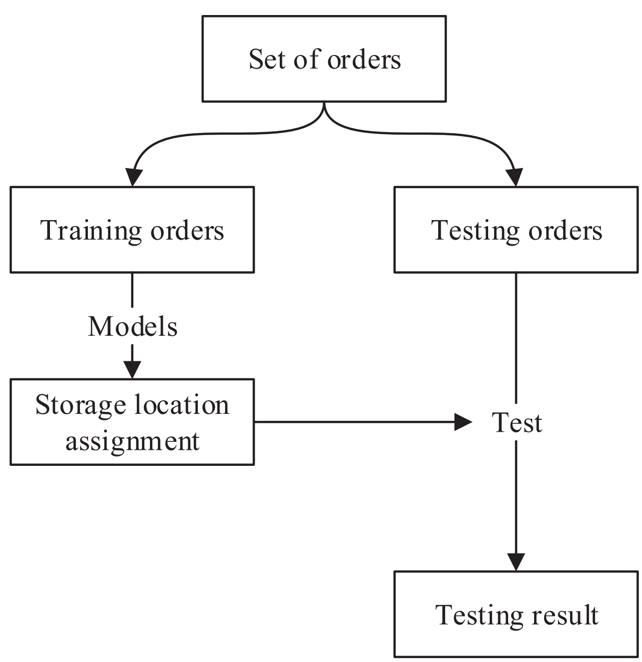

Fig. 2. Training-testing evaluation mechanism.

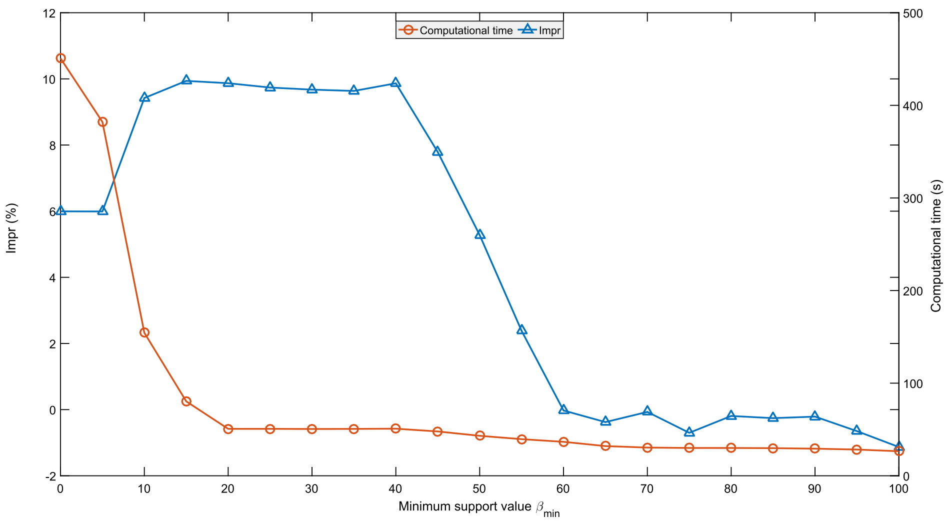

Fig. 3. Performance for different β_{min}.

where $TR(\cdot)$ refers to the TR of the given model.

Minimum support count. The minimum support count β_{min} indicates the minimum frequency of DCPs, i.e., the DCP m is considered to be frequent if $\beta_m \geqslant \beta_{min}$. With an appropriate value of β_{min}, DCPs that describe the item correlation in the original data can be mined out via FP-max (Grahne & Zhu, 2003). A large β_{min} may not be necessarily a suitable choice because only strong correlations (high frequencies) are mined out, whereas weak correlations (low frequencies) are disregarded. However, if β_{min} is excessively low, then weak correlations are mined out even if some of the correlations are accidental. As a result, the computational effort may be high since an excessive number of DCPs will be considered. We determine the β_{min} value through a number of trials. Given an instance of $A = R = 20$ generated in Section 5.3 as an example, to determine the appropriate β_{min}, a number of trials are conducted where the value of β_{min} is changed from 0 to 100 with each step of 5. The results displayed in Fig. 3 show that the *Impr* increases when $5 \leqslant \beta_{min} \leqslant 15$ and sharply decreases when $\beta_{min} > 40$. Additionally, the computational time is high when $\beta_{min} \leqslant 10$ and remains almost unchanged when $\beta_{min} \geqslant 20$. We also observe that there exists a reasonable interval within which β_{min} produces better frequent itemsets. As indicated by certain experiments, we find that a suitable interval exists for almost all instance scales, the bounds of which are 20 and 50. Therefore, in the following experiments, β_{min} is determined by a number of trials within [20, 50] and the best result is adopted.

Methods for comparison. Three typical approaches for the SLAP are selected from the following studies: Minimum Delay Algorithm (MDA, Wutthisirisart et al., 2015), Association Seed-Based Heuristic (ASBH, Chiang et al., 2014) and Clustering-Assigning Heuristic (CAH, Xiao & Zheng, 2010). MDA is an algorithm for a two-phase model. In the first phase, an item sequence is obtained by modelling the SLAP as a linear replacement assignment problem. In the second phase, items are assigned to storage locations along the S-shape route. For solving the linear replacement assignment problem, the MDA is employed. In each iteration, the MDA assigns each item to the most remote place, calculates their delays within the linear placement, and selects the item with the minimum value. ASBH is another heuristic with the characteristic, that the natures of items' relationships (complementary, substitutive or independent) are determined and used for assigning (Chiang et al., 2014). Regarding each aisle, the most correlated item pair is initially assigned, and subsequently, the aisle is fulfilled with items that are most correlated with the items that are already located in the aisle. CAH is a typical clustering-assigning method, that first groups items by the Hungarian method and then modifies and sorts the groups. In the assigning phase, items within each group that are already ranked in non-decreasing order are assigned to storage locations in a way that minimizes the travel distance to visit all these locations following the S-shape route. According to the results of Xiao and Zheng (2010), the parameters of CAH, TL and TU, are set to 0.1 and 0.6, respectively.

5.2. Computational results on real data

The real data in this study is collected from an online retailer, which sells products on the Internet. The dataset includes a total of 19,718 orders collected in one year; the information is summarized in Table 3. The picking area contains 788 items. The distribution center has a rectangular shape with 800 storage locations consisting of 20 two-sided aisles and 20 storage locations per aisle side. The width between adjacent aisles and locations are 4 m and 1 m, respectively. The distribution center adopts the picker-to-parts order-picking system and the dedicated storage strategy (DSS), where items are assigned to locations according to their picking frequency. The S-shape heuristic is employed for routing pickers. In the picking process, several orders are usually batched as a pick-list and sent to human pickers, who will retrieve the requested products and scan the code bars with their radio frequency identification (RFID) terminals. Once a pick-list is completed, the retrieved items are split into orders (re-bin), packed and delivered to customers. In this study, we do not consider the batching operation and assume that all the orders are picked one by one. We divide the orders into four groups according to the four quarters of the year and apply the aforementioned methods to each group. Since ASBH and SA are algorithms with random factors, we run them 10 times and pick the best result. The computational results are reported in Table 4.

From Table 4, we observe that MIH significantly outperforms the other methods, especially on the data of the first quarter. ASBH and SA are the next best methods. ASBH performs well in the 2nd, 3rd and 4th quarters, but fails in the 1st quarter, with an average improvement of 2.77 %, which is slightly better than those of SA and CAH, 2.55 % and 2.30 %. SA outperforms CAH in three of four quarters. The performance of MDA is not ideal. Table 3 indicates that the order size (number of items in one order) substantially varies, which causes MDA that tends to

Table 3
Information about the real data.

Data	Order number	Max. item number per order	Min. item number per order	Avg. item number per order	Stdev. item number per order
1st quarter	3989	409	1	12.91	24.71
2nd quarter	4747	212	1	12.30	16.13
3rd quarter	5032	209	1	13.84	19.93
4th quarter	5950	465	1	15.13	27.00

Table 4
Computational results for real data.

	DSS	MDA		CAH		ASBH		MIH		SA	
Data	Dist[a]	Dist	%	Dist	%	Dist	%	Dist	%	Dist	%
1st quarter	784,356	788,084	−0.48	753,092	3.99	802,392	−2.30	686,676	12.45	767,128	2.20
2nd quarter	998,854	997,882	0.10	979,644	1.92	957,692	4.12	922,802	7.61	965,958	3.29
3rd quarter	1,090,862	1,093,838	−0.27	1,079,672	1.03	1,034,358	5.18	1,004,418	7.92	1,065,464	2.33
4th quarter	1,352,928	1,352,198	0.05	1,322,316	2.26	1,297,650	4.09	1,246,902	7.84	1,320,588	2.39
Avg.	–	–	−0.15	–	2.30	–	2.77	–	8.96	–	2.55

[a] Dist refers to the travel distance for picking all orders in one quarter. Unit: meter.

assign items of large (small) order to locations far from (near to) the I/O point to perform worse. With the exception of MIH, we also observe that the improvement of the remaining methods over DSS is not significant. This finding may be explained by the weak item correlation in the dataset.

5.3. Numerical simulation

The real data in Section 5.2 presents weak correlation among items. In this subsection, we examine the proposed methods in the scenarios, where items are highly correlated. The situation can usually be found in production warehouses, where parts, components or materials that belong to the same BOMs or products are frequently retrieved together, and therefore, are highly correlated. We also examine how the proposed methods perform on different problem scales and correlation degrees. We generate instances of different scales, and then present and discuss the computational results.

5.3.1. Instance generation

The method for generating instances is similar to that of Xiao and Zheng (2010). We examine the proposed heuristics for different warehouse shapes determined by the aisle number A and the row number R. We assume that the number of items is equal to the number of storage locations, i.e., $N = L = 2AR$. This is without loss of generality, since dummy items requested by no order can be created, when $N < L$. In Xiao and Zheng (2010), the authors evaluate their multi-stage heuristic in a warehouse of $A = 10$, $R = 20$. In this study, we expand the problem scale and generate the numerical instance sets $S_{A,R}$ as follows:

- $A = 10, 20, 30$;
- $R = 10, 20, 30$.

There are $3 \times 3 = 9$ combinations. Therefore, we have 9 instance sets as follows: $S_{10,10}$, $S_{10,20}$, $S_{10,30}$, $S_{20,10}$, $S_{20,20}$, $S_{20,30}$, $S_{30,10}$, $S_{30,20}$, and $S_{30,30}$. For each set, 10 instances are generated, yielding a total of $9 \times 10 = 90$ instances. Regarding the other warehouse parameters, we set $v = 1$ and $h = 4$.

For each instance, a number of orders are randomly generated, in which several items are correlated. We use the concept of *common itemset*, a small subset of $\mathcal{N}$ (set of items), to reflect the demand dependency, as in Xiao and Zheng (2010). To make items correlated to each other, a number of itemsets are randomly generated and placed into each order with a probability of $\theta \in U[0, 1]$. θ is referred to as the *correlation degree* and $U[\cdot,\cdot]$ the uniform distribution. We set the order and itemset quantities proportional to N (cardinality of $\mathcal{N}$) to adapt to different problem scales. The involved parameters are generated as follows.

- Number of orders: $N \times U[30, 50]$;
- Number of common itemset: $N \times U[0.1, 0.5]$;
- Size of an order: $U[5, 15]$;
- Size of common itemset: $U[2, 5]$;
- Correlation degree: $\theta = 0.7$.

Note that we set $\theta = 0.7$ to ensure that the items are highly correlated. New instances with different θ are generated and solved in Section 5.3.3. For each order, the size is first determined and a decimal within [0, 1] is randomly generated to indicate whether it should include a few common itemsets. If the decimal is less (greater) than θ, then one to three (zero) common itemsets are randomly picked and added to the order. If the order is fulfilled, then stop and proceed to produce the next order; otherwise, the order is filled with items that are randomly picked from $\mathcal{N}$, set of items. Note that an item can appear more than once in an order, given that items can be appear with other items or independently.

5.3.2. Performance for different instance scales

We solve the generated instances by the previously mentioned methods, i.e., MDA, CAH, ASBH, MIH and SA. With respect to each set $S_{A,R}$, the improvements of the five methods over DSS for all instances

Table 5
Computational results with respect to different scales.

Instance	Avg.*Impr* (%)				CPU (s)					
set	MDA	CAH	ASBH	MIH	SA	MDA	CAH	ASBH	MIH	SA
$S_{10,10}$	3.50	6.71	7.62	7.72	9.02	0.03	0.13	0.16	0.01	2.08
$S_{10,20}$	4.88	7.18	8.74	7.50	10.89	0.10	0.45	0.77	0.04	7.99
$S_{10,30}$	6.51	8.94	11.46	8.81	12.55	0.22	1.02	2.85	0.07	21.99
$S_{20,10}$	1.16	6.32	7.34	7.33	8.27	0.09	0.43	0.72	0.03	6.04
$S_{20,20}$	4.82	7.29	9.66	7.65	10.26	0.46	2.08	7.65	0.14	47.02
$S_{20,30}$	4.81	8.99	12.45	8.94	12.51	1.35	4.82	28.33	0.28	132.47
$S_{30,10}$	2.60	5.13	5.45	6.07	6.83	0.23	1.10	2.91	0.08	19.59
$S_{30,20}$	3.42	8.27	10.93	8.77	11.25	1.06	4.63	27.19	0.25	94.36
$S_{30,30}$	4.15	9.28	12.75	9.22	13.16	3.04	11.35	92.72	0.53	335.15
Avg.	3.98	7.57	9.60	8.00	10.53	0.73	2.89	18.14	0.16	74.08

are calculated; the average improvements (Avg. *Impr*) are reported in Table 5.

From Table 5, we observe that SA significantly outperforms other methods for all cases. The second best method is ASBH; its good performance may be attributed to its characteristics of identifying and using item relationships. MIH is the third best method with an average improvement of 8.00 %, which is slightly better than that of CAH, 7.57 %. The improvement of MDA is still limited. The computational time is also reported in Table 5, which indicates that MIH runs very quickly; the computational time is less than one second for all instance scales. For SA, the running time is longer.

The results presented in Table 5 also indicate that with the given number of aisles A, the improvement significantly grows with an increase in the number of rows R. This phenomenon indicates that the warehouses that have more storage locations in one aisle would be more eager to be improved by considering item correlation. This fact is also observed by fixing the number of items N $(=2AR)$.

5.3.3. Performance for different correlation degrees

The instances in Section 5.3.2 are generated by assuming that items are highly correlated ($\theta = 0.7$). We further randomly generate a set of instances with different correlation degrees and re-evaluate the performances of the five methods. We fix the instance scale to $S_{20,20}$ and change the value of θ from 0 to 1 with each step of 0.1. For each value of θ, 10 instances are generated, and there is a total of 110 instances. These instances are solved using the five methods, and the average improvements (Avg. *Impr*) over DSS for each θ is reported in Fig. 4.

In Fig. 4, $\theta = 0$ indicates minimal correlation among items. In this case, no significant improvement over DSS is observed, and the performances of the five methods are almost equal. When items are weakly correlated, i.e., $\theta \in [0.1, 0.4]$, ASBH outperforms other methods. When the item correlation grows strong, i.e., $\theta \in [0.5, 1]$, SA obtains the best performance. We conclude that SA is especially useful in situations in which items are highly correlated and that ASBH is the best choice when item correlation is weak to medium. The performance of MIH is better than that of CAH and MDA for all θ values.

6. Conclusions

Order-picking is considered to be the most time- and labor-consuming operation in picker-to-parts warehouses; it accounts for a large proportion of the operating expense in a warehouse. This study aims at improving the order-picking efficiency by assigning storage locations to appropriate items, where the correlation among items is considered. Different from most studies that utilize picking frequency or pairwise correlation to make assignment decisions, we introduce the demand correlation pattern (DCP) to describe the item correlation and formulate the storage location assignment problem as an integer programming model. To solve the problem, the minimum increment heuristic (MIH) and simulated annealing (SA) algorithm are developed. The two methods are evaluated against extant methods using both real data and numerical instances. The computational results indicate the following:

- In general, the MIH is a good choice. It outputs competent assignments within a short computational time for all instance scales and correlation degrees. The MIH is especially useful for situations in which items are weakly correlated and order size considerably varies.
- When items are highly correlated, SA yields very competitive solutions compared with extant methods. Its performance decreases when the item correlation is weak.
- It is more effective to apply correlation-based storage strategies to warehouses with more storage locations per aisle.
- When items are independently demanded, the advantages of correlation-based strategies disappear and it is a good choice to apply the picking frequency-based strategy.

In this paper, we use DCP to describe item correlation and construct our model. The maximum frequent itemsets are determined by applying data-mining techniques to historical orders to determine DCPs. This is a good choice when item correlation is strong but may not be promising when the demand correlation is weak. These findings have been demonstrated in the computational experiments. Another limitation of our model and methods is that they are based on the S-shape route. In practice, however, other routing strategies are also usually adopted. For sake of simplicity, we also assume that only one I/O point exists in the warehouse. Multiple I/O points and separated input and output points are common, which need further investigation. Moreover, this paper does not consider aisle congestion, which makes influences to order-picking efficiency when the high demand and correlated items are assigned to locations near the I/O point.

This paper develops the correlation-based storage assignments for low-level picker-to-parts warehouses. Our model can be extended to the situation in which multilevel storage racks are employed. In this scenario, one storage rack can be occupied by different item types. The

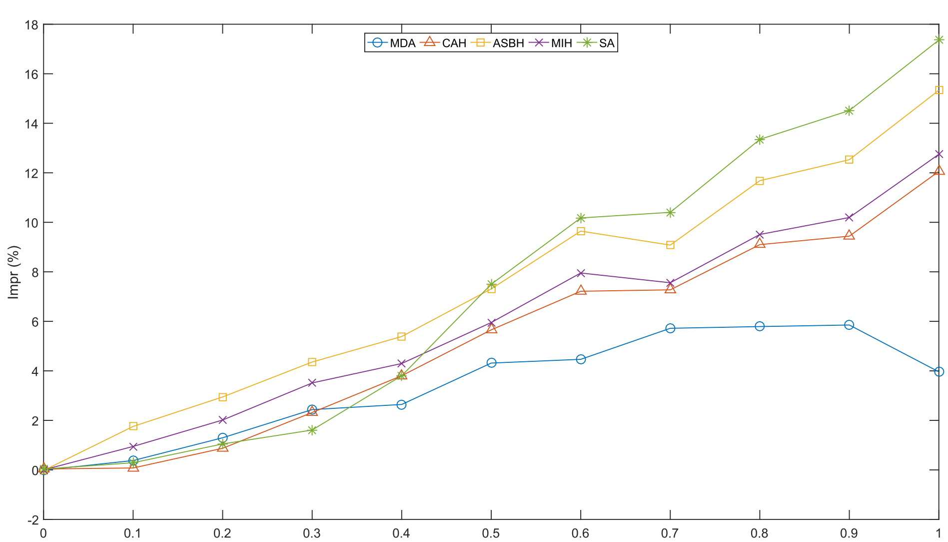

Fig. 4. Computational results for different correlation degrees.

vertical assignment decision should be made, and additional item properties, such as volume, weight or fragility, must be considered. For example, heavy items should be placed in low level locations and popular items should be placed in medium level locations for easier access by humans. Since the picker-to-parts order-picking system is manually operated, taking human factors into consideration is also a promising topic.

Acknowledgement

The work is supported by the National Natural Science Foundation of China with grant No. 71571006 and 71502015. The authors thank the anonymous referees for their encouraging comments and suggestions for improving the quality of the paper.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.cie.2019.01.027.

References

Bartholdi, J. J., & Gue, K. R. (2000). Reducing labor costs in an LTL crossdocking terminal. *Operations Research, 48*, 823–832.
Battini, D., Glock, C. H., Grosse, E. H., Persona, A., & Sgarbossa, F. (2016). Human energy expenditure in order picking storage assignment: a bi-objective method. *Computers & Industrial Engineering, 94*, 147–157.
Bindi, F., Manzini, R., Pareschi, A., & Regattieri, A. (2009). Similarity-based storage allocation rules in an order picking system: an application to the food service industry. *International Journal of Logistics Research and Applications, 12*, 233–247.
Boysen, N., De Koster, R., & Weidinger, F. (2018). Warehousing in the e-commerce era: A survey. *European Journal of Operational Research* in press.
Boze, Y. A., & Aldarondo, F. J. (2018). A simulation-based comparison of two goods-to-person order picking systems in an online retail setting. *International Journal of Production Research, 56*, 3838–3858.
Brynzér, H., & Johansson, M. I. (1996). Storage location assignment: Using the product structure to reduce order picking times. *International Journal of Production Economics,* 595–603.
Chan, F. T. S., & Chan, H. K. (2011). Improving the productivity of order picking of a manual-pick and multi-level rack distribution warehouse through the implementation of class-based storage. *Expert Systems with Applications, 38*, 2686–2700.
Chiang, D. M. H., Lin, C. P., & Chen, M. C. (2011). The adaptive approach for storage assignment by mining data of warehouse management system for distribution centres. *Enterprise Information Systems, 5*, 219–234.
Chiang, D. M.-H., Lin, C.-P., & Chen, M.-C. (2014). Data mining based storage assignment heuristics for travel distance reduction. *Expert Systems, 31*, 81–90.
Chuang, Y.-F., Lee, H.-T., & Lai, Y.-C. (2012). Item-associated cluster assignment model on storage allocation problems. *Computers & Industrial Engineering, 63*, 1171–1177.
Dallari, F., Marchet, G., & Melacini, M. (2009). Design of order picking system. *The International Journal of Advanced Manufacturing Technology, 42*, 1–12. https://doi.org/10.1007/s00170-008-1571-9.
De Koster, R., Le-Duc, T., & Roodbergen, K. J. (2007). Design and control of warehouse order picking: A literature review. *European Journal of Operational Research, 182*, 481–501.
Dijkstra, A. S., & Roodbergen, K. J. (2017). Exact route-length formulas and a storage location assignment heuristic for picker-to-parts warehouses. *Transportation Research Part E Logistics & Transportation Review, 102*, 38–59.
Elbert, R. M., Franzke, T., Glock, C. H., & Grosse, E. H. (2017). The effects of human behavior on the efficiency of routing policies in order picking: The case of route deviations. *Computers & Industrial Engineering, 111*, 537–551.
Frazelle, E. A. (2002). *World-class warehousing and material handling.* New York: McGraw-Hill.
Frazelle, E. A., & Sharp, G. P. (1989). Correlated assignment strategy can improve any order-picking operation. *Industrial Engineering, 21*, 33–37.
Gils, T.v., Ramaekers, K., Caris, A., & De Koster, R. B. (2018). Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review. *European Journal of Operational Research, 267*, 1–15.
Glock, C. H., & Grosse, E. H. (2012). Storage policies and order picking strategies in u-shaped order-picking systems with a movable base. *International Journal of Production Research, 50*, 4344–4357.
Glock, C. H., Grosse, E. H., Elbert, R. M., & Franzke, T. (2017). Maverick picking: The impact of modifications in work schedules on manual order picking processes. *International Journal of Production Research, 55*, 6344–6360.
Grahne, G., & Zhu, J. (2003). High performance mining of maximal frequent itemsets. *Proc. of the 6th SIAM Int1 workshop on High Performance Data Mining (HPDM 2003)* (pp. 135–143). .
Grosse, E. H., Glock, C. H., & Ballester-Ripoll, R. (2014). A simulated annealing approach for the joint order batching and order picker routing problem with weight restrictions. *International Journal of Operations and Quantitative Management, 20*, 65–83.
Grosse, E. H., Glock, C. H., & Jaber, M. Y. (2013). The effect of worker learning and forgetting on storage reassignment decisions in order picking systems. *Computers & Industrial Engineering, 66*, 653–662.
Grosse, E. H., Glock, C. H., Jaber, M. Y., & Neumann, W. P. (2015). Incorporating human factors in order picking planning models: framework and research opportunities. *International Journal of Production Research, 53*, 695–717.
Grosse, E. H., Glock, C. H., & Neumann, W. P. (2016). Human factors in order picking: A content analysis of the literature. *International Journal of Production Research, 55*, 1260–1276.
Hall, R. W. (1993). Distance approximation for routing manual pickers in a warehouse. *IIE Transactions, 25*, 76–87.
Hausman, W. H., Schwarz, L. B., & Graves, S. C. (1976). Optimal storage assignment in automatic warehousing systems. *Management Science, 22*, 629–638.
Heskett, J. (1963). Cube-per-order index – a key to warehouse stock location. *Transportation and Distribution Management, 3*, 27–31.
Hua, W., & Zhou, C. (2008). Clusters and filling-curve-based storage assignment in a circuit board assembly kitting area. *IIE Transactions, 40*, 569–585.
Jane, C.-C., & Laih, Y.-W. (2005). A clustering algorithm for item assignment in a synchronized zone order picking system. *European Journal of Operational Research, 166*, 489–496.
Kim, K. H. (1993). A joint determination of storage locations and space requirements for correlated items in a miniload automated storage-retrieval system. *International Journal of Production Research, 31*, 2649–2659.
Lai, K. K., Xue, J., & Zhang, G. (2002). Layout design for a paper reel warehouse: A two-stage heuristic approach. *International Journal of Production Economics, 75*, 231–243.
Larco, J. A., De Koster, R., Roodbergen, K. J., & Dul, J. (2017). Managing warehouse efficiency and worker discomfort through enhanced storage assignment decisions. *International Journal of Production Research, 55*, 6407–6422.
Lee, M.-K. (1992). A storage assignment policy in a man-on-board automated storage/retrieval system. *International Journal of Production Research, 30*, 2281–2292.
Li, J., Moghaddam, M., & Nof, S. Y. (2015). Dynamic storage assignment with product affinity and abc classificationa case study. *The International Journal of Advanced Manufacturing Technology, 84*, 2179–2194.
Liu, C.-M. (1999). Clustering techniques for stock location and order-picking in a distribution center. *Computers & Operations Research, 26*, 989–1002.
Liu, C.-M. (2004). Optimal storage layout and order picking for warehousing. *International Journal of Operations Research, 1*, 37–46.
Malmborg, C. J., & Bhaskaran, K. (1990). A revised proof of optimality for the cube-per-order index rule for stored item location. *Applied Mathematical Modelling, 14*, 87–95.
Marchet, G., Melacini, M., & Perotti, S. (2015). Investigating order picking system adoption: A case-study-based approach. *International Journal of Logistics: Research and Applications, 18*, 82–98 Case study.
Matusiak, M., De Koster, R., Kroon, L., & Saarinen, J. (2014). A fast simulated annealing method for batching precedence-constrained customer orders in a warehouse. *European Journal of Operational Research, 236*, 968–977.
Muppani, V. R., & Adil, G. K. (2008). Efficient formation of storage classes for warehouse storage location assignment: A simulated annealing approach. *Omega, 36*, 609–618.
Napolitano, M. (2012). 2012 Warehouse/DC operations survey: Mixed signals. *Modern Materials Handling, 51*, 48–56.
Otto, A., Boysen, N., Scholl, A., & Walter, R. (2017). Ergonomic workplace design in the fast pick area. *OR Spectrum, 39*, 945–975.
Pang, K.-W., & Chan, H.-L. (2017). Data mining-based algorithm for storage location assignment in a randomised warehouse. *International Journal of Production Research, 55*, 4035–4052.
Petersen, C. G., & Aase, G. (2004). A comparison of picking, storage, and routing policies in manual order picking. *International Journal of Production Economics, 92*, 11–19.
Petersen, C. G., & Schmenner, R. W. (1999). An evaluation of routing and volume – based storage policies in an order picking operation. *Decision Sciences, 30*, 481–501.
Petersen, C. G., Siu, C., & Heiser, D. R. (2005). Improving order picking performance utilizing slotting and golden zone storage. *International Journal of Operations & Production Management, 25*, 997–1012.
Ratliff, H. D., & Rosenthal, A. S. (1983). Order-picking in a rectangular warehouse: A solvable case of the traveling salesman problem. *Operations Research, 31*, 507–521.
Richards, G. (2014). *Warehouse management: A complete guide to improving efficiency and minimizing costs in the modern warehouse.* London: Kogan Page.
Roodbergen, K. J. R., & Vis, I. F. (2009). A survey of literature on automated storage and retrieval systems. *European Journal of Operational Research, 194*, 343–362.
Ruben, R. A., & Jacobs, F. R. (1999). Batch construction heuristics and storage assignment strategies for walk/ride and pick systems. *Management Science, 45*, 575–596.
Scholz, A., Henn, S., Stuhlmann, M., & Wäscher, G. (2016). A new mathematical programming formulation for the single-picker routing problem. *European Journal of Operational Research, 253*, 68–84.
Wutthisirisart, P., Noble, J. S., & Chang, C. A. (2015). A two-phased heuristic for relation-based item location. *Computers & Industrial Engineering, 82*, 94–102.
Xiao, J., & Zheng, L. (2010). A correlated storage location assignment problem in a single-block-multi-aisles warehouse considering BOM information. *International Journal of Production Research, 48*, 1321–1338.