International Journal of Production Research
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tprs20

A correlated storage location assignment problem in a single-block-multi-aisles warehouse considering BOM information

Jian Xiao [a] & Li Zheng [a]
[a] Department of Industrial Engineering , Tsinghua University , Beijing, China
Published online: 22 Jan 2009.

International Journal of Production Research
Vol. 48, No. 5, 1 March 2010, 1321–1338

A correlated storage location assignment problem in a single-block-multi-aisles warehouse considering BOM information

Jian Xiao* and Li Zheng

Department of Industrial Engineering, Tsinghua University, Beijing, China

(*Received 5 January 2008; final version received 10 October 2008*)

This paper deals with a correlated storage location assignment problem by considering the production bill of material (BOM) information. Due to the large number of parts in a BOM, the picking capacity constraint is considered. A mathematical model is formulated and a multi-stage heuristic is proposed. The heuristic relaxes the close interrelationships within the original problem through an improvement algorithm and an iterated approach to incorporate the effect of BOM splitting. In order to evaluate the performance of the heuristic, numerical experimentation is conducted in a single-block-multi-aisles warehouse, by using a randomly generated data set.

Keywords: storage assignment; BOM structure; order picking; dedicated storage strategy

1. Introduction

In production (including manufacturing and/or assembly) warehouses, parts have to be picked from specified storage locations. This process, known as the order picking process, is driven by work orders, each of which consists of a number of order lines. Each order line represents one part that has to be delivered to the work shop in a certain quantity, the order line quantity. The quantity is mainly determined by the production plan and the bill of material (BOM).

The picking process is, in general, the most laborious of all warehouse processes. It may consume about 60% of all labour activities in the warehouse (Ruben and Jacobs 1999). While the warehouse performance is typically judged upon throughput-based criteria, it is worthwhile to try to minimise the picking effort. In spite of the differences in material handling devices, there are mainly four methods used to reduce pick travel distances, namely: (1) determining good order-picking routes; (2) zoning the warehouse; (3) assigning orders to batches; and (4) assigning products to correct storage locations (Chen and Wu 2005). Discussion on these research areas can be found in some survey literature by, e.g., Gibson and Sharp (1992), Rouwenhorst *et al.* (2000), De Koster *et al.* (2007), Gu *et al.* (2007), and Muppani and Adil (2008).

*Corresponding author. Email: xiaojian99@mails.tsinghua.edu.cn

ISSN 0020–7543 print/ISSN 1366–588X online

DOI: 10.1080/00207540802555736
http://www.informaworld.com

The storage location assignment problem (SLAP), which is the main focus of this paper, is to determine a way of assigning items on a storage rack so that picking efficiency is maximised. Related factors in determining storage locations are the order-picking method, turnover rates, the demand dependence between products and storage space requirements (Kim 1993). The most popular storage location assignment strategies/policies include *dedicated storage*, *random storage* and *class-based storage* (Frazelle 1990). Under the *dedicated storage strategy*, each item has its own and fixed storage location. A *random storage* assigns the locations randomly for the items. *Class-based storage* can be seen as a trade-off between these two strategies, which partitions all the items into several classes, assigning each class a reserved region, within which the random storage is employed. Further information about storage assignment strategies can be found in the literature reviewed by De Koster *et al*. (2007). Dedicated storage strategy is considered in this paper.

The issues in this study are to determine the storage locations in a production warehouse, where there are limited and deterministic product BOMs. Furthermore, the production rate is relatively stable especially in a make-to-stock factory.

This study considers the entire BOM structure rather than each individual part alone, so it is a correlated assignment problem. The correlated assignment is different from the assignment based on cube-per-order index (COI) rule (Heskktte 1963), which is the optimal strategy when orders contain only a single item but does not always work well if items are correlated in order requests. The concept of correlated assignments for item storage is that, the more items with demand dependence are stored together, the greater becomes the chance of reduction in the routing length for a given list of orders. It is applied not only to the case of the part-to-picker systems (e.g., carousels) but also the picker-to-part systems.

Frazelle (1990) firstly describes the concept of correlated assignments for item storage where two items with relatively larger correlation value are stored together. He formulated SLAP as a NP-hard integer programming problem, and a two-phase construction heuristic for solving SLAP was designed. Amirhosseini and Sharp (1996) proposed an order satisfying correlation measure (OSCM) to measure how likely the two products are to satisfy the demand of orders in which they appear. They also proposed a clustering method that merges the attributes from the cluster with the new product added to it so at each stage the original cluster is nested inside the new one.

Since an order usually consists of more than two items, and thus has to be picked from multiple locations, new approaches are needed. Lee (1992) presented a heuristic procedure to identify those items with higher propensity, and then assign them closely in the storage rack following a space-filling curve. Liu (1999) gave a correlation measure for products based on how often the products appear together versus the maximal order size where products appear together. Kim (1993) and Sadiq (1993) studied the correlated storage assignment considering the warehouse operating costs, e.g., inventory-related costs and material handling costs, in the cluster formation.

These studies are mostly related to distribution centres or retail warehouses, where the types of order can be enormous and changing dynamically. So these studies usually use a similarity coefficient to measure the correlations between pairs of items, by statistical analysis in a sampling period. While in production warehouses, the order structures are limited and deterministic, which correspond to the product BOMs. Also, the production rate can be treated as the order frequency, which is relatively stable. So there should be better approaches to cluster items.

Brynzer and Johansson (1996) proposed a class-based storage assignment strategy emanating from the product structure by forming variant groups (VGs) from components with the same codification. Hsieh and Tsai (2001) developed a BOM oriented class-based storage assignment method by comparing the storage-bay attribute code and the material storage attribute code in the CIMS database. Neither of the studies mathematically formulated their problems.

Hua (2001) investigated cluster-based storage policies for the kitting area in a manufacturing environment. A correlation measure based on the percentage of orders containing both products is employed. Then a genetic algorithm is used to determine the clusters so that the total travel distance is minimised if the picker visits every SKU on a BOM in one trip.

In the existing literature, the storage assignment problem under dedicated storage policy based on the BOM structure is still not fully considered. Furthermore, they do not consider the picking capacity constraint when solving the correlated assignment problem. In our study, however, production is always conducted in batch quantity, and a product is always composed of many parts, it is impossible to pick the complete BOM in simply one route due to the picking capacity constraint. Thus the parts in a BOM should be split into several groups/sub-BOMs, and then each sub-BOM can be picked in one route.

The problem is defined and formulated mathematically in Section 2. In Section 3, a multi-stage heuristic is described to determine both the sub-BOMs for each BOM and the storage location for each part. This is followed by a numerical example and the result of numerical experimentation is provided. The paper concludes in Section 5 with a short discussion.

2. Problem formulation

In the production work shop, a product is not manufactured or assembled simply in one work station. There are always several work stations or work steps composing a production process. Different work stations need different groups of parts in the BOM, thus the parts have to be sorted to an individual work station before delivering to the spot. This operation is out of the scope of this study.

In production, there are multiple products/BOMs, each of which contains multiple parts. Different BOMs can contain the same parts, which is very common especially in platform-based products like cars. A single-block-multi-aisles warehouse like the one shown in Figure 1 is assumed to be capable of holding all the parts. The picking aisles are two-sided and are wide enough for two-way travel. The structure of the considered warehouse is limited to a two-dimensional storage allocation.

We assume that no stock splitting or mixture is allowed. In other words, each kind of item occupies exactly one location. The space in any location is large enough to keep each individual kind of item. A continuous review inventory system is assumed and no stock-outs occur. In addition, the replenishment of items in the storage area is not modelled. Since items are replenished in bulk quantities, the effect of this activity is minimal when compared to order picking (Ruben and Jacobs 1999). The picking capacity in one route is assumed to be large enough for at least one order line of any BOM aligned with the specified production batch quantity. Thus the splitting is

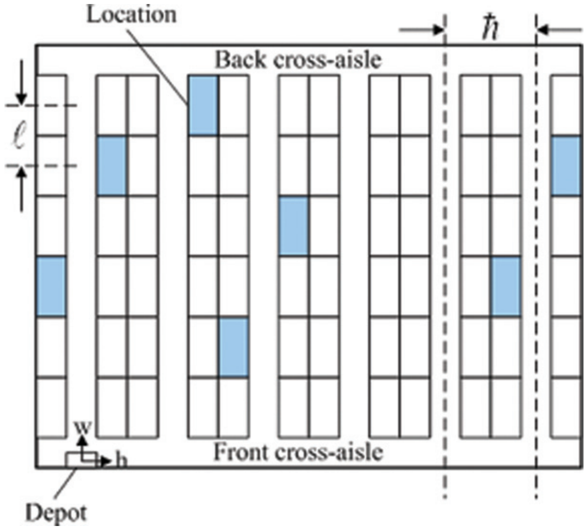

Figure 1. A single-block-multi-aisles warehouse layout.

prevented and each part in the BOM/sub-BOM can be picked in one route at most. Furthermore, we assume there is only one picker, thus congestion is out of consideration.

We use the following notation in this paper:

P = set of parts, the number of parts is denoted as N^P;
L = set of storage locations including the depot δ, which is treated as a virtual storage location that cannot store part; the number of locations is denoted as N^L. Without loss of generality, let $N^L = N^P + 1$;
B = set of BOMs, the number of BOMs is denoted as N^B;
C = picking capacity in one route, measured by e.g. volume;
q_b = production batch quantity of BOM b;
f_b = frequency of BOM b;
β_{pb} = 1 if part $p \in P$ is included in BOM b;
u_{pb} = quantity of part p in BOM b; naturally only when $u_{pb} > 0$, $\beta_{pb} = 1$;
c_p = capacity consumed by one unit of part p;
(w_l, h_l) = coordinates of a location $l \in L$; $(0, 0)$ for the depot δ;
ℓ = distance between any two adjacent locations;
$\hbar$ = distance between any two adjacent aisles;
K = maximum number of routes/sub-BOMs per BOM, intuitively $K \leq N^P$;
S_b = set of routes/sub-BOMs for BOM b, $|S_b| = K$;
d_s^b = travel distance to pick all parts of sub-BOM $s \in S_b$.

The variables are:

x_{ps}^b = 1 if part p is included in sub-BOM $s \in S_b$; 0 otherwise;
y_{pl} = if part p is assigned to location l; 0 otherwise;
z_{kls}^b = 1 if location l is visited immediately after k in sub-BOM $s \in S_b$; 0 otherwise.

Mathematically, the problem is then to determine values for all the variables that minimise:

$$\sum_{b\in B}\left(f_b \sum_{s\in S_b} d_s^b\right) = \sum_{b\in B}\left(f_b \sum_{k\in L}\sum_{l\in L} d_{kl} \sum_{s\in S_b} z_{kls}^b\right),$$

subject to:

$$\sum_{k\in L}\sum_{s\in S_b} z_{kls}^b = 1, \quad \forall l,\ \forall b \tag{1}$$

$$\sum_{l\in L}\sum_{s\in S_b} z_{kls}^b = 1, \quad \forall k,\ \forall b \tag{2}$$

$$\sum_{k\in L} z_{kms}^b - \sum_{l\in L} z_{mls}^b = 0, \quad \forall m \in L,\ \forall b,\ \forall s \tag{3}$$

$$z_{kls}^b \le \sum_{p_1\in P}\sum_{p_2\in P} x_{p_1 s}^b y_{p_1 k} x_{p_2 s}^b y_{p_2 l}, \quad \forall k,\ l \in L-\{\delta\},\ \forall b,\ \forall s \tag{4}$$

$$\sum_{k\in L}\sum_{l\in L} z_{kls}^b = \sum_{p\in P} x_{ps}^b + 1, \quad \forall b,\ \forall s \tag{5}$$

$$\sum_{p\in P} q_b u_{pb} x_{ps}^b c_p \le C, \quad \forall b,\ \forall s \tag{6}$$

$$\sum_{l\in L-\{\delta\}} z_{\delta ls}^b \le 1, \quad \forall b,\ \forall s \tag{7}$$

$$\sum_{k\in L-\{\delta\}} z_{k\delta s}^b \le 1, \quad \forall b,\ \forall s \tag{8}$$

$$\tau_k - \tau_l + \sum_{p\in P} \beta_{pb} \sum_{s\in S_b} z_{kls}^b \le \sum_{p\in P} \beta_{pb} - 1, \quad \forall k \ne l \in L-\{\delta\},\ \forall b \tag{9}$$

$$\sum_{p\in P} y_{pl} = 1, \quad \forall l \in L-\{\delta\} \tag{10}$$

$$\sum_{l\in L-\{\delta\}} y_{pl} = 1, \quad \forall p \tag{11}$$

$$\sum_{s\in S_b} x_{ps}^b = \beta_{pb}, \quad \forall b,\ \forall p \tag{12}$$

τ_k, τ_l arbitrary.

Travel distance between any two locations can be expressed as:

$$d_{kl} = \begin{cases} \min[\ell \cdot (w_k + w_l) + \bar{h} \cdot |h_k - h_l|] & h_k \neq h_l \\ \ell \cdot |w_k - w_l| & h_k = h_l \end{cases}, \quad \text{for } k, l \in L, k \neq l$$
$$d_{kk} = \Delta, \quad \text{for } k \in L,$$

where Δ is a sufficiently large positive value.

The first two constraints (1) and (2) represent that each stop in a sub-BOM tour, including the depot, has exactly one inlet stop and one outlet stop. Constraint (3) states that if a route visits a location, it must also depart from it. Constraint (4) ensures that, except for the depot, a sub-BOM picking tour only visits those locations that stores parts in the sub-BOM. Constraint (5) represents the relationship of the number of locations visited in a sub-BOM tour including the depot, and the number of parts in the sub-BOM.

Constraint (6) is the capacity constraint. It indicates that the maximum volume of parts that can be picked in one route cannot exceed the picking capacity. Constraints (7) and (8) express that there can be 'NULL' routes/sub-BOMs. But if one sub-BOM is not null, it can only be picked once. Constraint (9) is the sub-tour elimination constraint, derived from the travelling salesman problem (TSP) by Miller *et al.* (1960). In other words, the optimal solution must only have one tour connecting all the points in the sub-BOM, including the depot. Constraints (10) and (11) make sure each part is assigned to a single location except for the depot, and vice versa. Finally, constraint (12) ensures each part in a BOM will be put into exactly one sub-BOM.

3. A multi-stage heuristic algorithm

In order to solve the problem formulated in the previous section, there are three interrelated sub-problems to be considered:

(1) The assignment problem of allocating each part to a storage location.
(2) The partition problem of splitting each BOM into some sub-BOMs based on picking capacity constraint. For a product containing k parts, there are $2^k - 1$ possible clusters. Since it is unreasonable to consider all clusters explicitly except for very small values of k, we must develop other methods to cluster parts into sub-BOMs.
(3) The routing problem to pick each sub-BOM. The routing of order pickers in a warehouse is a special case of the travelling salesman problem (TSP) in which travel is restricted to aisles (Hall 1993). The problem classifies as a Steiner-TSP because of the two facts that some of the nodes do not have to be visited and that the other nodes can be visited more than once (De Koster *et al.* 2007). The difficulty with the Steiner-TSP is that it is in general not solvable in polynomial time.

Typical warehouse systems may have tens of thousands of parts, and several hundreds of BOMs, each containing several thousands of parts. Thus the formulation has millions of constraints and millions of variables or even more. Hence, a direct solution does not seem promising.

It is even worse than that. Let us state the decision version for the special case of this problem where: (1) every BOM contains exactly two parts; (2) the picking capacity is large enough that no splitting is required; and (3) there is only one aisle which indicates no complex routing to construct. The storage assignment problem under this situation will be reduced to the SLAP formulated by Frazelle (1990), which is NP-Hard.

Finding a joint optimal solution for three sub-problems is not a realistic approach, at least not for problem instances of the size encountered in practice. So it is more practical to devise some heuristic procedures which are fast enough to be useful, and whose solutions are good, but not necessarily optimal.

The objective of this section is to develop a simple heuristic algorithm to solve the joint problem. Our theory is to decompose the joint problem into three sub-problems, each of which relaxes some constraints in the original problem. The heuristics process is depicted in Figure 2.

The first problem is the part grouping and sequencing problem which relax the picking capacity constraints and the routing problem. The second problem is a bin-packing problem which divides the parts in each BOM into sub-BOMs according to the picking capacity constraints, without considering the routing problem. The grouping information derived from the first problem is used to improve the packing results. By solving this problem, various sub-BOMs are generated, each of which is a picking order. In order to obtain better part grouping and sequencing, the first problem is solved again by taking the sub-BOMs and their related information as inputs. This iteration can be conducted as many times as possible until a stable state is reached. The third problem is to assign each part to a specific location according to the part sequencing information by considering the specific routing method or simply by following some 'space-filling curve' as described by Lee (1992).

Actually the iterations between the first problem and the second to reach a stable state can be very time consuming. In this paper, only a one-iteration approach is considered. The complete heuristic composes four stages, as depicted in Figure 2.

3.1 *Stage 1: preliminary part clustering and sorting*

The clustering operation is to generate similar groups of parts such that the parts within a group are more strongly related to each other than those in different groups. Cluster analysis is composed of many diverse techniques for recognising structure in a complex data set. This technique is employed by most of the previous literature on clustering items for correlated storage location assignment, e.g., Frazelle (1990), Kim (1993) and Liu (1999). While in our study, the BOMs are predefined and in most cases are constant, the remaining work is to find a suitable clustering algorithm.

Group technology is an important scientific principle in improving the productivity of manufacturing systems. The application of group technology to manufacturing starts with finding the families of similar parts and forming the associated groups of machines

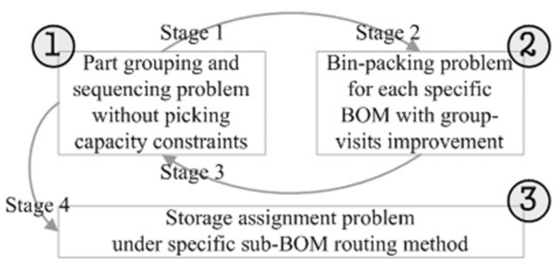

Figure 2. Decomposition of the problem.

(Seifoddini and Wolfe 1986). This process is referred to as 'machine-component grouping' or 'machine-part group formation' problem (MGFP). This problem has a very similar background with our problem and Lee (1992) did apply a modified MGFP algorithm to identify item groups. However, his definition of similarity coefficient can be problematic, because it depends largely on the absolute value of order frequency and not the relative value. For example, by changing order frequency vector from [10 20 30 40] and [0.1 0.2 0.3 0.4] the similarity coefficient will be very different. In this study, a modified version of his algorithm is proposed as follows:

Step 0: Construct the part-BOM incidence matrix $A_{N^P \times N^B} = (a_{pb})$, $p \in P$, $b \in B$ where $a_{pb} = 1$ if part p is included in b and otherwise 0.

Step 1: Build the part similarity coefficient matrix $S_{N^P \times N^P} = (s_{ij})$ with:

$$\begin{cases} s_{ij} = \sum_{b \in B} f_b(a_{ib} \& a_{jb}) / \sum_{b \in B} f_b(a_{ib} | a_{jb}) & i \neq j \\ s_{ii} = -\Delta \end{cases},$$

where Δ is a sufficiently large positive number. $\sum_{b \in B} F_b(a_{ib} \& a_{jb})$ indicates the frequency of BOMs that contain both part i and j, and $\sum_{b \in B} F_b(a_{ib} | a_{jb})$ indicates the frequency of BOMs that contain either part i or j, where & is the logical *AND* operator, and | is the logical *OR* operator.

Step 2: Solve the 0–1 assignment problem with the cost matrix S, the decisions variables Y_{ij}, $i, j = 1, \ldots, n$, and the objective of maximising the total assignment cost. As in the TSP, allocation in any diagonal element is to be avoided and the objective is maximisation, the diagonal elements are forced to be a negative number.

Step 3: Identify all part groups G_k, each comprising the elements $\{i, j, k, \ldots, p, q\}$ that make $Y_{ij} = Y_{jk} = \cdots = Y_{pq} = Y_{qi} = 1$.

Step 4: For each group, eliminate any item whose similarity coefficient with every other element in the group is less than a threshold value TL. The eliminated one itself becomes a new group. Repeat this process until no further elimination is possible.

Step 5: Update the similarity coefficient GS_{kl} between any pair of groups G_k and G_l by taking:

$$\max_{i \in G_k, j \in G_l} \{s_{ij}\}.$$

Step 6: Merge any pair of groups whose GS is larger than a threshold value TU. If one group appears in more than one pair to be merged, the pair with largest similarity is merged (if there is more than one largest, merge any one pair randomly). Then go to step 5 until no more groups can be merged, go to step 7.

Step 7: Compute the frequency of each part $f_p = \sum_{b \in B} a_{pb} f_b / c_p$ for $p \in G_k$, $k = 1, \ldots, N^G$, or in other words for $p \in P$. Then sort the parts within each group in non-decreasing order of frequency. If there are more than two parts with the same frequency, sort them randomly.

Step 8: Set N^G = the number of groups formed and N_k^G = the number of parts in group G_k. Compute the frequency of each group $F_k = (\sum_{p \in G_k} \sum_{b \in B} a_{pb} f_b / c_p) / N_k^G$ for $k = 1, \ldots, N^G$, then sort the groups in non-decreasing order of frequency. If there are more than two groups with the same frequency, sort these groups according to the largest part frequency of each individual group; while it is still the same, then the second largest etc.

3.2. Stage 2: BOM splitting

In this stage, due to the fact that batch picking of different BOMs is not allowed, there is no need to consider relations between different BOMs, so each BOM is treated individually.

Effective BOM splitting should consider both the location of parts in the BOM and the picking capacity requirements of each part. The latter is predefined but the former is the decision variable and remains unknown until this stage. However, the parts clustering information from last stage can be a good substitution.

The problem of splitting a BOM into sub-BOMs is equal to grouping the parts in the BOM into batches. If the locations information is not considered, this problem reduces to the bin-packing problem, which is NP-complete. In this context, the bin-packing problem is to determine the minimum number of sub-BOMs necessary to accommodate all the parts in a BOM. A number of efficient heuristics have been proposed to solve the bin-packing problem. One of the best is the first fit decreasing (FFD) algorithm despite a little bit of runtime overhead (Ruben and Jacobs 1999). By considering parts clustering information, we propose a modified FFD algorithm with a 2-exchange improvement method as follows:

Step 0: Sort the list of parts within BOM b in non-increasing order of capacity consumed, which equals $q_b u_{pb} c_p$. Note that the total number of sub-BOMs is not pre-determined but determined dynamically.

Step 1: Create a single sub-BOM which includes only the first part on the sorted list.

Step 2: Proceed down the sorted list of parts, the sub-BOM membership of each part is determined by scanning the list of previously created sub-BOMs from the first to the latest one until one is found to accommodate the part without exceeding the picking capacity constraint. A part that cannot be accommodated in any previously created sub-BOM is assigned to a new sub-BOM and the sub-BOM list is extended by one.

Step 3: Repeat step 2 until all parts are assigned to a sub-BOM.

Step 4: Set $M_b =$ the number of sub-BOMs formed and compute the total number of groups (from stage 1) visited to pick all the sub-BOMs as:

$$\Gamma = \sum_{i=1}^{M_b} \tau_i,$$

where τ_i is the number of groups visited to pick the ith sub-BOM. Note that, as all sub-BOMs share the same production batch quantity, it is eliminated from the equation.

Step 5: If the value of Γ can be reduced by exchanging the assignment of any two parts without exceeding the capacity constraint, go to step 6. Otherwise, stop.

Step 6: Exchange the corresponding assignment, and go to step 4.

Each BOM is treated one by one following this heuristic until a list of sub-BOMs for all BOMs is generated.

3.3 *Stage 3: refined part clustering and sorting*

The sub-BOMs generated in the last stage are the actual picking orders. In order to obtain better clustering results, the same heuristic as stage 1 is employed to generate a new set of groups by considering the sub-BOMs information. At the end of this stage, all the parts are sorted in a non-decreasing order. Let N' be the number of groups formed, G'_k, $k = 1, \ldots, N'$, be the kth group and n'_k be the number of parts in G'_k.

3.4 *Stage 4: storage location assignment and picking route generation*

The next thing to be determined is how to assign the items to storage locations. The easiest approach may be the closest-location assignment rule, in which the remaining closest location from the I/O point is allocated preferentially to the remaining outranked part. As mentioned previously, this assignment scheme may yield a near optimal result in a single-location picking system. However, for our study, which is a multi-location picking system, this rule may not be a good choice. Regarding this, we will suggest another location sequencing rule adapted to different picking routing methods.

The routing problem is a Steiner-TSP, which is an NP problem. However, for the type of warehouse in our study, it was shown by Ratliff and Rosenthal (1983) that an algorithm does exist that can solve the problem in running time which is linear to the number of aisles and the number of pick locations.

However, in industrial practice, the routing problem is mainly solved by heuristics because there are some disadvantages of optimal routing in practice. For example, an optimal algorithm is not available for every layout. Even if there is, the optimal routes may seem illogical or non-intuitive to the pickers who then may waste time on confirming the specified routes.

De Koster *et al.* (2007) reviewed several heuristic methods for routing order pickers in single-block warehouses. Petersen (1997) carried out a number of numerical experiments to compare six routing methods: the *S-shape*, *return*, *largest gap*, *mid-point*, *composite* and *optimal* in a situation with random storage. He concluded that a best heuristic solution is on average 5% over the optimal solution. As routing is not the focus of our study, we choose only two of them for illustration purposes, which are the S-shape method and the largest gap method as depicted in Figure 3.

The *S-shape* (or *traversal*) *algorithm* (Hall 1993) is one of the simplest heuristics for routing order pickers. When the *S-shape algorithm* is employed, any aisle containing at least one pick is traversed entirely (except the last visited aisle). Aisles without picks are not entered. From the last visited aisle, the order picker returns to the depot. This algorithm is efficient, when entering and leaving aisles is time-consuming, such as in the case of high-bay order picking trucks (De Koster *et al.* 2007).

In cases where entering aisles is not time consuming and the density of pick items per aisle is low, the *largest-gap algorithm* (Hall 1993) may outperform the *S-shape algorithm*. In this strategy, all aisles except the first and last visited are left at the same side as they were entered. The order picker returns at the position of the largest gap between two adjacent locations to be visited in the aisle (including aisle end points). A gap represents

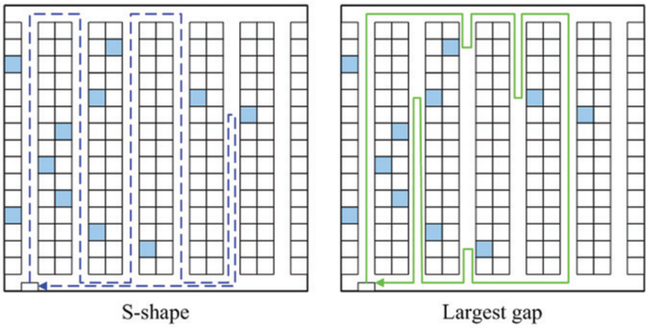

Figure 3. S-shape and largest gap method.

the separation between any two adjacent picks, between the first pick and the front aisle, or between the last pick and the back aisle. The largest gap is the portion of the aisle that the picker does not traverse. If the largest gap is between two adjacent picks, the order picker performs a return route from both ends of the aisle. Otherwise, a return route from either the front or back aisle is used. The largest gap within an aisle is therefore the portion of the aisle that the order picker does not traverse. The back aisle can only be accessed through either the first or last aisle.

The location assignment heuristic procedure is as follows:

Step 0: Select a routing algorithm, in this study, S-shape or largest-gap method. Mark the status of all locations in the warehouse as *available*. Let $k = 1$.

Step 1: Find n'_k locations from the *available* locations in a way that minimises the travel distance to visit all these locations following the specific routing algorithm. As the parts in G'_k are already ranked in non-decreasing order, they are assigned to the locations from the first to the last following the visit order.

Step 2: Mark all the locations taken in step 1 as *unavailable*. Let $k = k + 1$.

Step 3: If $k \leq N'$, go to step 1; otherwise stop.

4. Numerical example

To investigate the performance of the heuristic, experiments with a relatively large scale data set was conducted. The experimental warehouse configuration is depicted in Figure 4 with 20×20 locations and 10 aisles, where $\ell = 1$ and $\hbar = 3$. Other parameters include:

Number of parts:	400;
Picking capacity:	25000;
Number of BOMs:	$U(30, 50)$;
BOMs frequency:	$U(1, 10)$;
Batch quantity of BOMs:	$10 \times U(1, 10)$;
Capacity consumed per unit of part:	$U(1, 50)$;
Size (number of parts) of BOMs:	$U(50, 70)$;
Quantity of parts in BOMs:	$U(1, 5)$;

where $U(\cdot, \cdot)$ indicates the discrete uniform distribution.

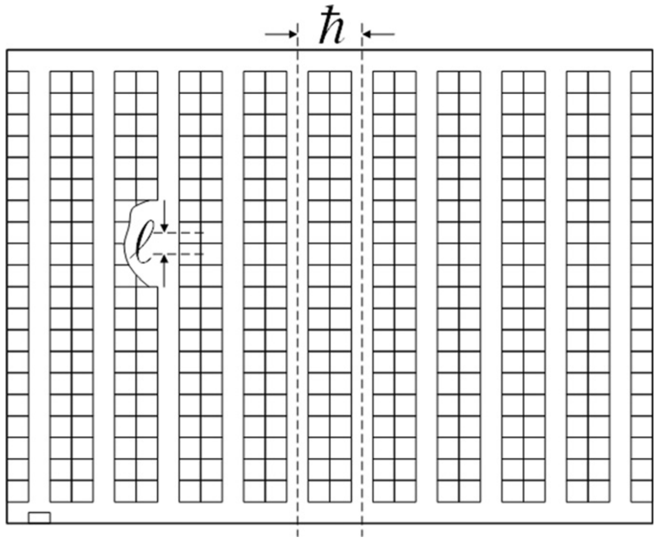

Figure 4. Experimental warehouse layout.

If there is no subjective judgment in advance, each BOM structure can be generated randomly. But in industrial practice, different products may always share the same platform or at least same assemblies. In order to reflect this situation in the experimental data, we generate some special part-groups and make them to be more likely to appear in the BOMs. These special part-groups include the platform part-groups and the commonly used part-groups, which is generated according to:

Number of platform part-groups:	$U(3, 5)$
Size of each platform part-group:	$U(10, 15)$
Number of commonly used part-groups:	$U(8, 20)$
Size of each commonly used part-group:	$U(5, 10)$

Then use the following approach to generate the parts for each BOM:

```
For each experimental problem
     Randomly generate the special part-groups.
     Randomly generate the number of BOMs, denoted as N^B.
  For each BOM b
       Randomly generate the size of BOM b, denoted as N_b^B.
       Set all parts status to available.
       Randomly pick exact one platform part-group into BOM b.
       Set all the parts in the part-group as unavailable.
       Randomly pick U(2, 5) commonly used part-groups into BOM b.
       Set all the parts in the part-groups as unavailable.
       While the number of parts in BOM b is less than N_b^B
          Randomly select a part from the available parts into BOM b.
          Set the status of the part to unavailable.
       End While
     Next
  Next
```

Table 1. Comparison of heuristics under different *TU*.

	S-shape				Largest gap				
TU	1S	2S	3S	4S	1L	2L	3L	4L	Avg
0.2	−5.51	−5.51	0.04	0.03	−12.18	−12.11	0.04	0.03	−4.40
0.3	5.29	5.26	0.12	0.08	−2.26	−2.01	0.10	0.08	0.83
0.4	7.61	7.43	1.26	0.56	1.90	1.57	−0.18	−0.60	2.44
0.5	7.57	7.47	6.39	5.51	2.29	2.16	4.67	4.15	5.03
0.6	**7.66**	7.48	**7.66**	**5.90**	**3.04**	2.37	**5.79**	**5.03**	**5.62**
0.7	7.55	7.52	6.22	4.54	2.66	**2.53**	5.40	4.64	5.13
0.8	7.56	**7.52**	4.36	2.86	2.99	2.52	5.29	4.65	4.72
0.9	7.28	7.50	1.93	0.56	2.76	2.51	4.33	3.55	3.80

In order to find out the relation of the heuristic efficiency with the threshold value, experiments under different *TL* and *TU* value are conducted and the results are compared. To keep the comparability, 300 problems are randomly generated and are the same data set for each experiment. These experiments are separated into two groups, in each of which the following five different heuristics are compared:

(1) The complete 4-stage heuristic.
(2) The heuristic without conducting 2-exchange improvement in stage 2
(3) The heuristic without conducting stage 3.
(4) The heuristic without conducting 2-exchange improvement in stage 2 and without stage 3
(5) COI-rule assignment strategy

In the first group of experiments, *TL* is set to 0.1 and *TU* is set to 0.2–0.9 in an increment step of 0.1, respectively. Table 1 lists the experimental results for both *S-shape* and *largest gap* routing method. The value in the 1S–4L columns represents the average of:

$$100(d_{\mathrm{COI}} - d_i)/d_{\mathrm{COI}}, \quad i = 1S, 2S, \ldots, 4L.$$

Even though the experiments are conducted using the same data set, a different threshold value can lead to very different grouping results. To illustrate how the heuristics behave with variation in different threshold value, we employ the following grouping index:

$$I = \frac{|G|}{|P|},$$

where $|G|$ is the number of groups generated in either stage 1 or stage 3, and $|P|$ is the number of all parts. The index indicates the degree of how the parts are grouped. As the value of grouping index approaches zero, the parts are becoming highly-grouped but the grouping index cannot be zero because the least number of groups is 1. The average grouping indexes for the problems under different *TU* are shown in Figure 5.

Using the S-shape routing method, the relation between the heuristic efficiency with the threshold value *TU* is depicted in Figure 6. The complete 4-stage heuristic outperforms others in most cases. But the 2-exchange improvement has a very slight impact on the efficiency. This is caused by the fact that stage 3 will re-group the parts while the

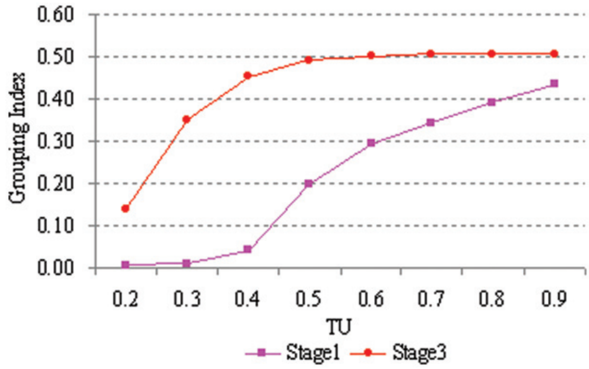

Figure 5. Grouping indexes under different *TU*.

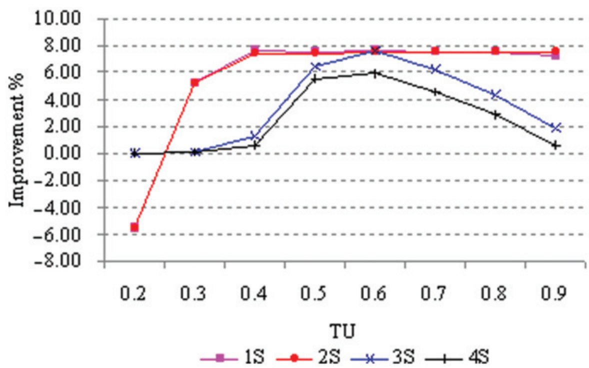

Figure 6. The heuristic efficiency under different *TU* (S-shape).

improvement is based on the group information in stage 1. The effect of 2-exchange improvement can only be noticed when stage 3 is not conducted, by comparing 4S to 3S.

The *TU* value is important for all four heuristics, despite their different characteristics. When the *TU* value is too small, say 0.2, they have a bad performance especially for 1S and 2S. As the *TU* value increases, the performance increases. It is interesting to find out that 1S and 2S have a very close relation to the grouping index of stage 3, where the growth tends to be very slight when *TU* reaches a specific value.

3S and 4S act in a very different way. In the first half part, say $TU \leq 0.6$, the increasing trend is similar to the grouping index of stage 1, but in the last half part, the performance drops down while the grouping index keeps increasing. The possible reason may be that, the groups generated in stage 1 are based on the original BOMs, but the final picking order is the sub-BOMs and are different from the original BOMs. Although the groups of stage 1 may be suitable for picking the '*entire*' original BOMs, it is not for the sub-BOMs. While 1S and 2S take the advantage of stage 3 that revises the part clustering to a certain extent, they perform more stably.

When the largest gap routing method is employed, the relation between the heuristic efficiency with the threshold value *TU* is depicted in Figure 7. The trend of L1/L2 and L3/L4 is similar to S1/S2 and S3/S4, respectively. The difference between Figure 7 and Figure 6 is, L3/L4 outperform L1/L2 in most cases. This is mainly due to the naïve storage

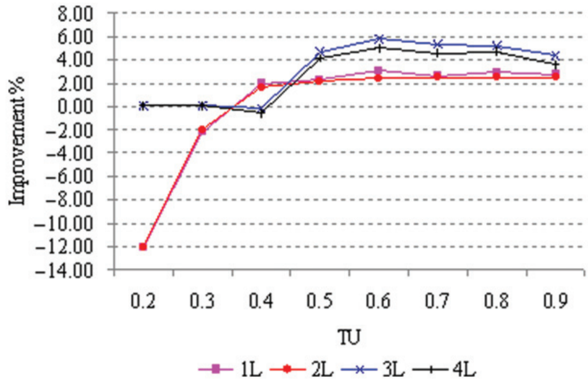

Figure 7. The heuristic efficiency under different *TU* (largest gap).

Table 2. Comparison of heuristics under different *TL*.

	S-shape				Largest gap				
TL	1S	2S	3S	4S	1L	2L	3L	4L	Avg
0.00	7.39	7.30	7.66	5.89	2.67	2.17	5.80	5.04	5.49
0.05	7.49	7.39	7.66	5.89	2.74	2.21	**5.81**	**5.04**	5.53
0.10	**7.66**	**7.48**	7.66	5.90	**3.04**	**2.37**	5.79	5.03	**5.62**
0.15	6.97	6.85	7.67	5.90	2.26	1.56	5.79	5.03	5.25
0.20	5.46	5.56	**7.69**	**5.93**	1.80	1.45	5.78	5.03	4.84

assignment strategy employed in stage 4, so a better assignment rule should be developed to best suit the largest gap routing method.

Table 1 shows that, the best average performance emerged when $TU = 0.6$. So in our next group of experiments, we fix *TU* to 0.6 and test the performance under different *TL* values. The same data set in the last experimental group is used, and the results are shown in Table 2. The average grouping indexes for the problems under different *TL* are shown in Figure 8. The relation of the heuristic efficiency with the threshold value *TL* is depicted in Figure 9, using both S-shape and largest gap routing methods.

It can be seen that, *TL* has slight impact on heuristics 3 and 4 no matter what routing method is employed. This is because *TL* has little effect on the grouping indexes of stage 1. Further inspection into the generated part groups, we find out that in stage 1, most groups generated by solving the 0-1 assignment problem are good groups, i.e., parts within each individual group have high similarities with each other, so the threshold value performs little function. This should be attributed to our BOM generation approach.

However, for heuristics 1 and 2, when *TL* is set too large, the performance will decrease rapidly. This is because there are much more smaller size sub-BOMs in stage 3 compared to the number of BOMs in stage 1, which depresses the similarity coefficients of most parts and makes the grouping index sensitive to *TL* and *TU*. A large *TL* will break down some actually good groups without forming better groups. A smaller *TU* may remedy this to a certain extent.

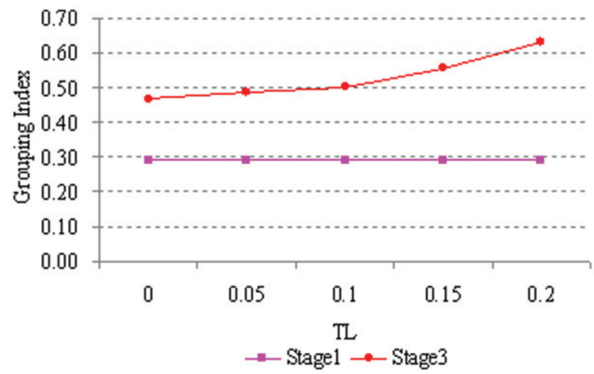

Figure 8. Grouping indexes under different *TL*.

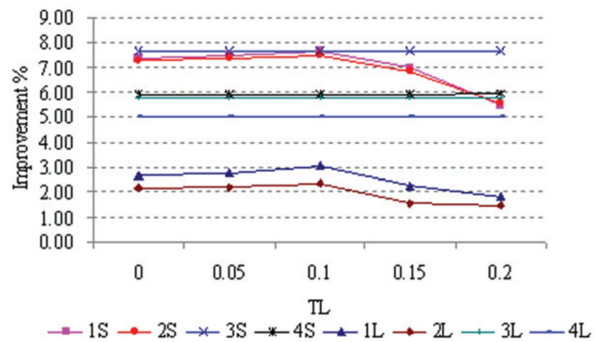

Figure 9. The heuristic efficiency under different *TL*.

5. Conclusion

In this paper, a mathematical model was presented and a multi-stage heuristic was proposed to solve a correlated storage location assignment problem in a single-block-multi-aisles warehouse for production. In a stable production environment, the parts relations are always constant based on the product BOMs. By discovering the BOM structure and frequency, *similar* parts can be clustered into groups and assigned to locations near to each other within the group. This can be a good approach to improve picking efficiency. However, production is always conducted in batch quantity, and a product includes many parts, so it is almost impossible to pick all of them in one route due to the picking capacity constraint. Thus, the parts in a BOM should be split into several sub-BOMs. Furthermore, a different routing strategy may be employed due to the unique warehouse layout and/or facilities. The routing strategy then can have great impact on the storage assignment order. All these reasons make the problem very complex and hard to find the optimal solution. We then proposed a practical heuristic to find a good but not optimal assignment. The heuristic relaxes the interleaving relations in the original problem and separates them into sub-problems. A one-iteration is conducted to incorporate the effect of BOM splitting. Actually the heuristic can be adapted to multi-iteration with a cost of computing time.

In the experiments, the complete 4-stage heuristic and three modified multi-stage heuristics were studied compared to the COI rule. Experimental results showed that, the layout obtained from the multi-stage heuristic is considerably better than that obtained by applying the COI rule but the threshold value can have a great influence on our heuristics, so it should be carefully chosen. It was also observed that, even with the same parts sequence, a different assignment and/or routing method can lead to very different picking efficiency. As mentioned in the previous section, the heuristic algorithm could be implemented with any other assignment rules and routing methods. Further study into the interrelations between various assignment rules and routing methods should be conducted.

Although our study focused on a warehouse layout with only one single block, the multi-stage heuristics process is suitable for other layouts. In our heuristics, the first three stages were employed to group and sort the parts and have no concern with the layout. Only in the last stage, the assignment and/or routing heuristics were layout specific. Considering a warehouse with multi blocks and more cross aisles, there exists the extended versions for both S-shape and largest gap heuristics (Roodbergen and De Koster 2001), which could be used in stage 4. However, the efficiency in this situation should be further inspected.

Acknowledgement

This research is supported by the National 863 plan of China (No. 2008AAA04Z102).

References

Amirhosseini, M.M. and Sharp, G.P., 1996. Simultaneous analysis of products and orders in storage assignment. *Manufacturing Science and Engineering*, MED-vol. 4 Proceedings of ASME International Conference on Manufacturing Science & Engineering (MSEC), Atlanta, GA, USA, 803–811.

Brynzer, H. and Johansson, M.I., 1996. Storage location assignment: using the product structure to reduce order picking times. *International Journal of Production Economics*, 46–47 (1), 595–603.

Chen, M.C. and Wu, H.P., 2005. An association based clustering approach to order batching considering customer demand patterns. *Omega*, 33 (4), 333–43.

De Koster, R., Le-Duc, T., and Roodbergen, K.J., 2007. Design and control of warehouse order picking: a literature review. *European Journal of Operations Research*, 182 (2), 481–501.

Frazelle, E.H., 1990. *Stock location assignment and order batching productivity*. Thesis (PhD). Georgia Institute of Technology, Atlanta, Georgia.

Gibson, D.R. and Sharp, G.P., 1992. Order batching procedures. *European Journal of Operations Research*, 58 (1), 57–67.

Gu, J., Goetschalckx, M., and McGinnis, L.F., 2007. Research on warehouse operation: a comprehensive review. *European Journal of Operations Research*, 117 (1), 1–21.

Hall, R.W., 1993. Distance approximation for routing manual pickers in a warehouse. *IIE Transactions*, 25 (4), 77–87.

Heskktte, J.L., 1963. Cube-per-order index: a key to warehouse stock location. *Transportation and Distribution Management*, 3 (4), 27–31.

Hsieh, S. and Tsai, K.C., 2001. A BOM oriented class-based storage assignment in an automated storage/retrieval system. *International Journal of Advanced Manufacturing Technology*, 17 (9), 683–691.

Hua, W., 2001. *Cluster based storage policies in kitting area*. Thesis (PhD). Georgia Institute of Technology, Atlanta, Georgia.

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 (11), 2649–2659.

Lee, M.K., 1992. A storage assignment policy in a man-on-board automated storage/retrieval system. *International Journal of Production Research*, 30 (10), 2281–2292.

Liu, C.M., 1999. Clustering techniques for stock location and order-picking in a distribution center. *Computers and Operations Research*, 26 (10), 989–1002.

Miller, C., Tucker, A.W., and Zemlin, R.A., 1960. Integer programming formulation of the traveling salesman problem. *Journal of the ACM*, 7 (4), 326–329.

Muppani, V.R. and Adil, G.K., 2008. Efficient formation of storage classes for warehouse storage location assignment: a simulated annealing approach. *Omega*, 36 (4), 609–618.

Petersen, C.G., 1997. An evaluation of order picking routing policies. *International Journal of Operations & Production Management*, 17 (11), 1098–1111.

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

Roodbergen, K.J. and De Koster, R., 2001. Routing methods for warehouses with multiple cross aisles. *International Journal of Production Research*, 39 (9), 1479–1504.

Rouwenhorst, B., *et al.*, 2000. Warehouse design and control: framework and literatures review. *European Journal of Operations Research*, 122 (3), 515–33.

Ruben, R.A. and Jacobs, F.R., 1999. Batch construction heuristics and storage assignment strategies for walk/ride and pick systems. *Management Science*, 45 (4), 575–596.

Sadiq, M., 1993. *A hybrid clustering algorithm for reconfiguration of dynamic order picking systems*. Thesis (PhD). University of Arkansas, Fayetteville, Arkansas.

Seifoddini, H. and Wolfe, P.M., 1986. Application of the similarity coefficient method in group technology. *IIE Transactions*, 18 (3), 271–277.