Constrained clustering method for class-based storage location assignment in warehouse

Chao-Lung Yang and Thi Phuong Quyen Nguyen
*Department of Industrial Management,
National Taiwan University of Science and Technology, Taipei, Taiwan*

Received 5 September 2015
Revised 27 October 2015
Accepted 19 December 2015

Abstract

Purpose – Class-based storage has been studied extensively and proved to be an efficient storage policy. However, few literature addressed how to cluster stuck items for class-based storage. The purpose of this paper is to develop a constrained clustering method integrated with principal component analysis (PCA) to meet the need of clustering stored items with the consideration of practical storage constraints.

Design/methodology/approach – In order to consider item characteristic and the associated storage restrictions, the must-link and cannot-link constraints were constructed to meet the storage requirement. The cube-per-order index (COI) which has been used for location assignment in class-based warehouse was analyzed by PCA. The proposed constrained clustering method utilizes the principal component loadings as item sub-group features to identify COI distribution of item sub-groups. The clustering results are then used for allocating storage by using the heuristic assignment model based on COI.

Findings – The clustering result showed that the proposed method was able to provide better compactness among item clusters. The simulated result also shows the new location assignment by the proposed method was able to improve the retrieval efficiency by 33 percent.

Practical implications – While number of items in warehouse is tremendously large, the human intervention on revealing storage constraints is going to be impossible. The developed method can be easily fit in to solve the problem no matter what the size of the data is.

Originality/value – The case study demonstrated an example of practical location assignment problem with constraints. This paper also sheds a light on developing a data clustering method which can be directly applied on solving the practical data analysis issues.

Keywords Principal component analysis, Class-based storage, Constrained hierarchical clustering, Warehouse location assignment

Paper type Case study

1. Introduction

Warehouse is a facility which stores items including work in process (WIP) for manufacturing and finished goods for shipping. An efficient warehouse management system must have the following features: save manpower, equipment and time; use the storage space efficiently; and maintain the good quality and quantity of the stored materials/goods (Frazelle, 2001). In order to maintain warehouse efficiency, determining the location assignment policy to store items is crucial for warehouse management.

Hong-Yu Industrial Company (HY), one of precision metal stamping and mold manufacturing company in Taiwan, has been applied dedicated storage policy for

The authors would like to thank reviewers' comments for improve this work. The authors also appreciate the financial support from National Science Council of Taiwan, ROC (Contract No. NSC 102-2622-E-011-009-CC3) and Hong-Yu Industrial Company. Particularly, authors thank the sharing of domain knowledge and storage requirement from Hong-Yu Industrial Company which provide practical managerial insights to contribute this research work.

Industrial Management & Data Systems
Vol. 116 No. 4, 2016
pp. 667-689

0263-5577
DOI 10.1108/IMDS-09-2015-0361

decades to store their fragile and sophisticated products and WIP in multiple warehouses around manufacturing sites. Basically, the dedicate policy assigns items to the pre-specified locations reserved based on prior planning. It is easy to maintain the storage and the material handling is not costly. Although the storage space might not be utilized efficiently because of the variation of retrieval and space used, it has been operated smoothly for decades when the warehouses were established with spacious floor plan.

However, since 2012, due to the business expansion, the warehouse system needs to contain more and more items and WIP, and the retrieval efficiency was going to be critical for production process. Without dramatically changing the facility layout, the management echelon initiated a project to reconsider the warehouse storage policy. Based on the survey and literature review, the random and class-based storage were reviewed. Although random storage policy which assigns items randomly based on the available slot can utilize space more efficiently, the considerable material handling is required because the location is not fixed. Because material handling of sophisticated stamping products is costly, class-based storage was considered. Basically, in class-based warehouse, all stored items are clustered to different classes based on some criteria and each class is assigned to a particular location. Within each block of location for an item class, the items in that class are stored randomly in the block. A popular criterion, Cube-per-order index (COI) which captures item's retrieval frequency and its required storage space was commonly suggested to optimally assign item classes to storage locations.

Although class-based storage has been studied and implemented extensively, few literatures addressed how to cluster the stock items for class-based storage. In addition, due to the special needs of operating or storing criteria, the storage constraints of item-to-location and item-to-item exist in warehouse. These constraints, in fact, confine the item clustering to consider not only the item similarity but also the operating constraints. For example, the item-to-location constraints mean the items have specified characteristics and should be stored in a particular location in the warehouse. Usually, fragile or sophisticated items should be placed at the lower racks or wrapped by protective materials in a warehouse. Some items which are easily corroded should be placed in the drying environment where the low humidity is maintained. Or, flammable items with oily coverage are better placed properly and avoided the fire or heat sources. In addition, the item-to-item constraints mean an item should not be placed together with another item. For instance, to reduce the retrieval error caused by human mistakes, the items which are easily confused with others by their appearances, functions, or material types are usually allocated in different areas. Obviously, the associated storage constraints should be considered in the item clustering to ensure that the stored items are kept in the appropriate conditions.

In this research, a constrained clustering method was proposed to group stuck items for a class-based warehouse by considering the constraints and special needs of stuck operation. The item-level must-link (ML) and cannot-link (CL) constraints were constructed to meet the operation and storage criteria among items. For each formed item sub-groups, the principal component (PC) loadings which provides the supplemental information describing COI distribution of the item sub-groups were used in the distance computation of the constrained clustering algorithm to better differentiate item sub-groups. After clustering stuck items, the heuristic algorithm for the location assignment model was developed to allocate the items of each cluster by considering COI and the item-to-location storage requirement.

In this work, the data set collected from one of HY warehouse containing the type of materials, shipping frequency, package size, and other information regarding the stock constraints was studied as a case study. The new storage allocation based on this research has been implemented on site to provide better efficiency of the warehouse management. This paper is structured as follow. Section 2 reviews the relevant research of warehouse management and constrained clustering method. Section 3 provides the data analysis framework of the constrained clustering method and heuristic location assignment for warehouse location assignment problem. In Section 4, the evaluation of the proposed method and the case study using the proposed method to solve the real-world location assignment problem are addressed. Finally, the conclusions, along with the future research directions are drawn in Section 5.

2. Literature review

2.1 Class-based storage location assignment in warehouse

Essentially, warehouse operations are governed by three types of storage policies: random, dedicated, and class-based storage (Hausman *et al.*, 1976). The class-based storage policy, which was first presented by Hausman *et al.*, allocates the items among the number of classes and keeps an area for each class within the storage space. The implementation of the class-based storage location problem was introduced more detail by Van Den Berg (1999), including in three stages: identifying the number of classes; assigning items to classes based on their demand rates; and allocating storage location for classes. There are several researchers have compared the performance of the class-based storage policies to random and dedicated storage (Eynan and Rosenblatt, 1994; Kouvelis and Papanicolaou, 1995; Muppani and Adil, 2008a, b; Petersen *et al.*, 2004; Sooksaksun *et al.*, 2012). Among these policies, the class-based storage is the most effective policy because it is integrated the advantages of random and dedicated storage policies (Sooksaksun *et al.*, 2012). The literature reviews of the location assignment with class-based storage policy are addressed as follows.

Van den Berg (1996) introduced elaborately to class-based storage policy in the single-command warehouse which was solved by using the dynamic programming approach (DPA). Larson *et al.* (1997) provided a three-phase heuristic procedure including identify storage space; assign material to storage medium; and allocation of floor space to maximize space utilization and minimize handling cost of the class-based storage warehouse. Muppani and Adil (2008b) first developed a non-linear integer programming model which minimizes the total pick travel distance to form the class and assign the storage location. Then the model was solved by using the branch and bound algorithm (BBA) (Muppani and Adil, 2008a). The objective in the class-based storage location assignment was extended to minimize the sum of storage space cost and handling cost which was solved by using BBA and simulated annealing (Muppani and Adil, 2008c). Vishkaei and Rashti (2011) deliberated the more reality problem as considering the total number of picks, storage level, and efficiency are the fuzzy number. The fuzzy model which attained two objective minimizing the sum of storage cost and handling cost; and maximizing the total efficiency was solved by the fuzzy DPA. Sooksaksun *et al.* (2012) proposed an approach that considered simultaneously the determination of aisle layout and storage location assignment. Then the problem was solved by using particle swarm optimization algorithm to minimize the average travel distance.

As addressed in some of above reviewed methods, the COI is an item-oriented index to assign the storage location to the item class. The COI of an item is defined as the ratio of the total storage locations required for an item to the number of retrievals performed per period. The COI-based assignment rule deliberates to list all items in ascending order of COI value and allocates the lowest of them to the most desirable location (Lai *et al.*, 2002). Rather than considering item-oriented assignment by COI, Frazelle and Sharp first proposed order-oriented strategy by clustering items based on their pairwise occurrence in a pick-up order (Lambert *et al.*, 1998). Mantel *et al.* (2007) also invented a new assignment strategy called order-oriented slotting which stores items close to each other in the dedicated storage warehouse if they appear together in orders. This order-oriented assignment in fact considers the pairwise correlated constraints which can be specified by ML constraints among items. Therefore, intuitively, the constrained clustering method can be applied on order-oriented storage for item clustering.

Although class-based storage has been studied extensively, relatively few literatures addressed how to identify the class boundaries of items. Based on author's knowledge, Rosenwein (1994) implemented cluster analysis approach to locate items within a warehouse. The clustering model was formed as a *p*-median problem to determine the group of items that had similar ordering pattern. The items in the same cluster was stored near to the others in the storage space. Liu (1999) used the clustering techniques to deal with the stock location and order picking in a distribution center. The clustering problem was formulated as an integer programming model to group item types and customers. Jane and Laih (2005) considered the clustering method for assigning items to a synchronized zone. The similarity between two items was measured based on co-appearance of items in the same order. Egas and Masel (2010) used the clustering technique to determine storage assignments based on the stock-keeping units (SKUs). The clustering model grouped the SKUs that were ordered together into the same cluster. Then the cluster with the most frequently ordered SKUs was assigned to the most desirable locations.

Based on previous works, we found that the relatively small numbers of item classes were determined for class-based storage. How to cluster or define the item class is not clear in literature. Especially, when the item-to-location or item-to-item storage constraints as mentioned earlier exist in the warehouse, how to define the item class (or group) to fulfill the constraints and also enhance the retrieval efficiency is unknown for location assignment. Therefore, in this research, the constrained clustering method was developed to cluster the items in warehouse into different classes/groups by considering item's properties and its associated constraints.

2.2 Review of constrained clustering method

In recent years, the semi-supervised clustering method emerges and attracts a lot of attention from the data mining community. In contrast to traditional (unsupervised) clustering, the semi-supervised clustering conducts the clustering process under the guidance of some supervisory information to improve efficiency and purity of clustering (Jiang *et al.*, 2013; Zhao *et al.*, 2012). The supervisory information can be represented by two kinds of instance constraints: ML constraint and CL constraint (Shaohong and Hau-San, 2009; Wagstaff and Cardie, 2000; Wagstaff *et al.*, 2001). A ML constraint indicates that two instances must be grouped in the same cluster. In contrary, a CL constraint specifies that two instances should not be placed in the same cluster. The ML and CL constraints for the storage items can be determined based on the material characteristics, storage restriction, and retrieval convenience, or by the expert in the warehouse domain.

When multiple items needs to be placed together based on pairwise ML constraints such as order-oriented strategy, the group-level ML constraints can be constructed.

In other words, it is a special case of the constrained clustering problem in which the r pre-existing partitions $(M_1, \ldots, M_r)$ among n items is pre-determined. Each data instance x_i in the data set D must be belonging to one of $(M_1, \ldots, M_r)$ partitions exclusively where $i = 1, \ldots, n$, n is the total number of data instances, and r is the number of pre-existing partitions or groups. Then, clustering problem is in fact clustering r pre-groups to k classes. In this research, the warehouse items have group-level ML constraints which restrict the items to be placed together or separated depending on the characteristics of items. Therefore, group-level ML constraints on warehouse item data set were implemented to construct the r groups of items which have similar characteristics.

For the constrained clustering method, there are two major methods were proposed in literature: constrained K-means and constrained hierarchical clustering algorithm. K-means method is one of the partitional clustering algorithm that try to find K non-overlapping clusters (MacQueen, 1967). This method uses the centroid to represent the cluster. Wagstaff *et al.* (2001) formed the constrained version of K-means, named COP-KMEANS, which aimed to find a partition that can balance the within-cluster variation with the number of unsatisfied instance-level constraints. The COP-KMEANS method deals with the constraints by calculating the transitive closure set and replacing all of them by using the centroid of all the points in that set. Although K-means based method is popular, the randomness of assigning initial cluster centers leads to stochastic process which means the clustering result and convergence vary depending on the initial centers. Therefore, in this research, the constrained hierarchical clustering method was applied to avoid the randomness.

Davidson and Ravi (2005a, b) first applied ML and CL constraints on agglomerative hierarchical clustering. Similar to constrained K-means, the constrained agglomerative clustering first handles ML constraints by applying the transitive closure. Then, for each iteration of assigning cluster member, the algorithm checks if the CL constraints is violated. This method identifies the number of cluster based on the generated dendogram that is a tree structure to represent the clustering result. Wagstaff and Cardie (2000) made a comparison between these two methods in terms of the input, output, the complexity of algorithm and the advantage and disadvantage. The constrained K-means and hierarchical algorithms can be significantly efficient when clustering data with ML and CL constraints. However, only using the centroids to represent the ML groups in performing clustering may not be very effective due to the slack of information of the original data set such as shape and distribution. Therefore, this research integrates the principal component analysis (PCA) to the original distance matrix which represents the ML groups instead of only using the centroids. More detailed information about integrating PCA with constrained clustering method will be addressed in the next section.

3. Methodology

In this research, a data framework which includes three stages using clustering method to group stuck items for the class-based storage location assignment problem was developed. The first stage is to collect and process warehouse data to reveal the constraints between items. Then the constrained clustering method is applied to cluster items or item groups by considering the location and storage constraints. Here, the agglomerative hierarchical clustering method is particularly used for storage item clustering. Because the tree structure, also called dendrogram, can be provided by hierarchical clustering, it helps on showing how and which storage items can be placed together and how they are merged for clustering. Comparing with other partition-based clustering such as *K*-means which has randomness of initialization, the agglomerative hierarchical clustering is more applicable for warehouse item clustering.

During the clustering process, the PCA information of combined retrieval frequency and required storage (COI domain) in each pre-determined item sub-groups is added for computing the distance to differentiate the item groups by their COI distribution. The clustering results will be used to assign the items to the storage location in the third stage. The heuristic location assignment algorithm considering COI information is integrated with the proposed constrained clustering method. Figure 1 shows the flowchart of the conceptual model.

3.1 Data processing

The warehouse contains various items which might have different characteristics to restrict the item assignment in the storage planning. Based on the literature review, we can construct the ML and CL constraints identified based on the characteristics of items and managerial requirements. The ML constraints among items can be constructed to gather the items by considering material type, storage restriction, or handling characteristics. The items which have similar characteristics such as the material types or material handling process can be pre-grouped to form the ML constraints to gather the items together. The pairwise notation x_1+x_2 can be used to denote that x_1 item should be clustered together with x_2 to form the ML constraint.

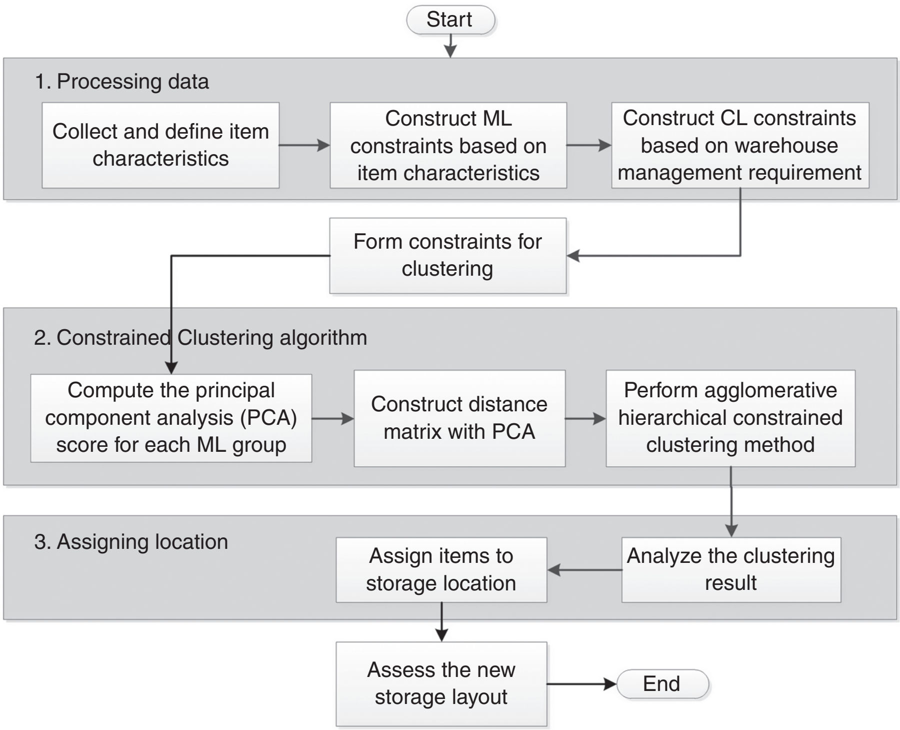


Figure 1.
The conceptual model of the constrained clustering method for location assignment problem

In fact, multiple ML constraints might construct multiple transitive closure sets. Suppose x_1+x_2 and x_2+x_3 exist, these two ML constraints also induce x_1+x_3 ML constraints and constructs x_1+x_2+x_3 set. Basically, once ML constraints are formed, the distance or similarity measures among ML constraints will be set to be 0 or relatively small number to enforce the items with ML constraints are clustered together.

Similar to ML constraints, the CL constraints can be constructed based on operating requirement. For example, in the case study, stuck items made by cold rolled steel material are not suggested to be placed together with items made by tinplate because of their confusing appearance. The pairwise CL constraint denoted by x_1−x_2 can formed to separate item x_1 to item x_2. While performing clustering, the distance measure between x_1 and x_2 will be set to be a relatively large number to enforce clustering process to merge x_1 and x_2 together. The CL constraints can be extended to separate a group of items from another group. Please note that if a CL x_1−x_2 and a ML x_2+x_3 exist, it also implies CL x_1−x_3 exists. The pseudo code ***imposeCannotLinks*** listed below can handle the distance matrix by considering ML and CL constraints together:

Algorithm 1. imposeCannotLinks.

imposeCannotLinks (distance matrix $\boldsymbol{D}$, constraints $\boldsymbol{ML}$, $\boldsymbol{CL}$)
 for $(\boldsymbol{i}, \boldsymbol{j}) \in \boldsymbol{CL}$ and $(\boldsymbol{j}, \boldsymbol{k}) \in \boldsymbol{ML}$
  $\boldsymbol{D}_{i,j}, \boldsymbol{D}_{i,k} = \boldsymbol{max}\,(\boldsymbol{D}) + 1$
 return $\boldsymbol{D}$

Based on the specifications of items under stamping process in HY warehouse, Table I lists the item groups which are specified by the item material, secondary processing,

Sub-group	Material	Secondary processing	Need dry environment/ heat treatment	Location constraints	No. of items
1	Stainless steel	WIP	Dry		16
2	Stainless steel	The finished product to be packaged	Heat		33
3	Cold rolled steel	WIP	Dry	Not with tinplate (group 7)	38
4	Copper-nickel alloy	WIP			12
5	Copper-nickel alloy	Product storage			13
6	Copper-nickel alloy	The finished product to be packaged			33
7	Tinplate	WIP	Dry	Not with cold rolled steel (group3)	10
8	Tinplate	Product storage			44
9	Tinplate	The finished product to be packaged			133
10	Brass	WIP	Dry		12
11	Brass	Be finished package			10
12	Beryllium copper	WIP	Dry		26
13	Aluminum	WIP	Dry		39
14	Phosphor bronze	WIP	Dry		16

Table I. The ML constraints constructed by item material category and type of work in process and the associated number of items in the group. The reason of constructing these item sub-groups is to clarify the location constraints which limit item groups' locations based on material properties and secondary processing. For example, the WIP and finished products need to be kept separated for packing and retrieval convenience. Therefore, the locations of WIP and finished products should be near to the manufacturing zone, and the packing area, respectively. In addition, the items made by certain material might need heat treatment or drying environment in the warehouse. The locations of the mentioned items will be limited on heat treatment and air conditioning chamber accordingly. In order to analyze the item groups, the ML constraints are formed to restrict the items with the same material and secondary processing in the same group. Table I lists the pre-determined sub-groups among the stored items which are constructed by ML constrained with their material type and secondary processing.

Also, as mentioned earlier, one CL constraints is specified in Table I to limit the WIP items made by cold rolled steel material to be placed together with the WIP items made by tinplate. Please note that ML and CL link constraints listed here are based on the practical specifications of stored items in the mentioned warehouse of this case study. For the larger warehouse with more items stored, more ML and CL constraints need to be considered and the complexity of clustering constraints are extended. When performing clustering on multiple transitive closures, all items in each transitive closure set (sub-group) will be replaced with the centroid of all items in that set. The clustering on original items will be transformed to the clustering on several centroids of sub-groups with items which has no ML constraints with others.

3.2 COI distributions of items sub-group

In order to diminish the effect of different scales of frequency and required space, the raw data were normalized and the minus of the normalized data are shifted to 0 to reflect the conventional COI index. Let:

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix} \tag{1}$$

is a n by p matrix with all n items and p variables. In order to reduce the effect of scale of variables and also maintain the positive values for representing COI, the normalization process is conducted on each variable of $\boldsymbol{X}$ to obtain the $\boldsymbol{R}$:

$$\boldsymbol{R}_j = \frac{\boldsymbol{X}_j - \overline{\boldsymbol{X}_j}}{\boldsymbol{\sigma}_{X_j}} - \min(\boldsymbol{X}_j), j = 1, \ \ldots, p \tag{2}$$

Figure 2 shows an example of item data where the retrieval frequency and required storage space of items are plotted. As can be seen in Figure 2, three sub-groups of items constructed by multiple ML constraints are identified as different symbols, and the centroids of each sub-group are specified with solid symbols. Obviously, these three sub-groups have unique data distribution. In order to specify the data distribution on retrieval frequency and required space domain (COI domain), the first PC of each sub-group are denoted by solid-line arrows and dashed arrows are for second PC. Based on

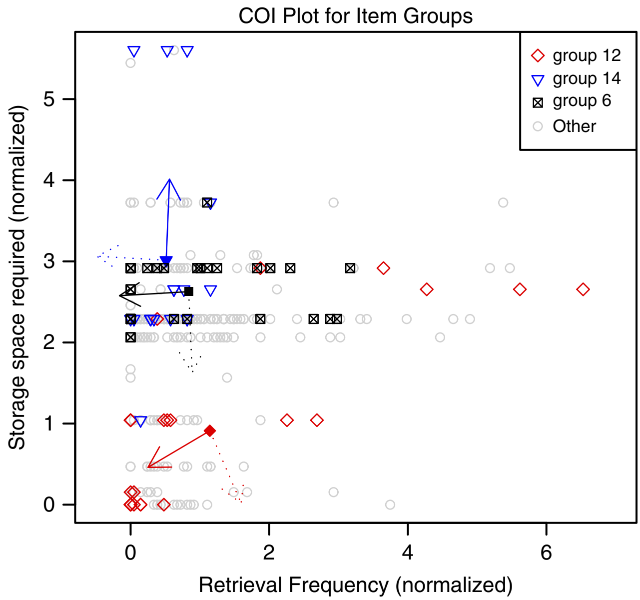


Figure 2.
Required storage location and retrieval frequency (COI) of stored items

definition of COI (required space/retrieval frequency), the items located as top-left corner in Figure 2 have relatively larger COI comparing with items in bottom-right corner.

In fact, item group 12 which has relatively low COI which means items in sub-group 12 are retrieved frequently and the space required are not large. The data points in sub-group 12 also show the linear distribution in terms of COI ratio. It means the items in sub-group 12 have relatively consistent COI ratio. The first PC of sub-group 12 simply specifies these consistency characteristics of the data distribution where items with larger required space have larger frequency correspondingly. On the contrary, sub-group 14 has larger COI and the direction of first PC is quite different from it of sub-group 12. It can be shown that items in sub-group 14 have larger variation of COI ratio. It might need special care on assigning items in warehouse due to the variation on required space. Practically, the items which need larger space should be allocated in consecutive locations for storage convenience. Similarly, items in sub-group 6 also have non-consistent COI, but the major variation is items' retrieval frequencies. In this case, items with high retrieval frequency should be assigned near the input/output (I/O) point for retrieval convenience.

Based on the data exploration on COI domain, we can conclude that not only COI ratio should be considered while assigning items of sub-group in the class-based warehouse, but also COI distribution of each sub-group should be taken care while allocating items in the warehouse. For the sub-group with not consistent COI distribution, the assigning algorithm needs to prioritize items in the sub-group by either the required space or frequency to further enhance the retrieval or storage convenience, respectively. In order to cluster items or item sub-groups by considering the COI distribution, in this research, the PCA loading data is suggested to be considered when computing the distance between data points or sub-groups. The item assignment also considers the COI distribution. The more detailed information about the proposed clustering method and assignment heuristic method are addressed in the following sections.

3.3 Constrained clustering algorithm with PCA

In order to cluster item groups by considering their storage characteristics without losing information caused by centroid replacement, this research integrated the constrained clustering method with PCA which aims to provide the supplemental information describing original item groups. PCA is a multivariate statistical analysis which is usually used for data dimension reduction while still keeps the information of the most important features of data set (Johnson and Wichern, 2002). The measurement of the amount of information transmitted by each PC is its variance. In this research, the items' PCA on COI domain is integrated with the hierarchical constrained clustering by adding the PCs loading in the centroid matrix of each item group as a supplementary element to interpret data.

The PCs loading is derived from the application of eigenvalues and eigenvectors. Suppose r sub-groups $\boldsymbol{R}_{ML1}, \ldots, \boldsymbol{R}_{MLr}$ are constructed by transitive closure ML constraints. The centroids $m_1, \ldots, m_r$ of each $\boldsymbol{R}_{ML1}, \ldots, \boldsymbol{R}_{MLr}$ are used to represent each sub-group during the clustering process. Because PCA is only applied on COI features which are retrieval frequency and required space indicated as *freq* and *space*, respectively, the PCs on *freq* and *space* can be identified as the two eigenvectors of the covariance matrix $Cov(R_c)_{freq,space}$ of a particular data partition (or sub-group) R_c:

$$Cov(R_c)_{freq,space} = \begin{bmatrix} Cov(\sigma_c)_{freq,freq} & Cov(\sigma_c)_{freq,space} \\ Cov(\sigma_c)_{freq,space} & Cov(\sigma_c)_{space,space} \end{bmatrix} \quad (3)$$

Two eigenvalues and two eigenvectors of $Cov(R_c)_{freq,space}$ can be constructed as in equation below. The more details about PCA computation can be found in Jlloffe (2005), Johnson and Wichern (2002) and Smith (2002):

$$\begin{aligned} eigenvalues(R_c) &= (\lambda_1, \lambda_2)_{R_c} \\ eigenvectors(R_c) &= (v_1, v_2)_{R_c} \end{aligned} \quad (4)$$

While dealing with the distance calculation in agglomerative clustering, the PC loadings which specify the distribution information about items sub-groups are added with centroid of each sub-group. The new distance calculation is defined as $\delta'(\boldsymbol{c_i}, \boldsymbol{c_j})$ where $\boldsymbol{c_i}$ and $\boldsymbol{c_j}$ are different clusters. As defined in equation below, for each $\boldsymbol{c_i}$, the centroids of $\boldsymbol{c_i}$ and the transpose of eigenvectors v_1, v_2 are appended together for distance calculation. This new distance calculation not only computes the centroids difference between two clusters, but also considers the similarity of PC loadings as well to determine which two clusters should be merged first:

$$\delta'(c_i, c_j) = \text{distance}\left(\left[m_i \text{ of } c_i + (v_1', v_2') \text{ of } c_i\right], \left[m_j \text{ of } c_j + (v_1', v_2') \text{ of } c_j\right]\right) \quad (5)$$

Algorithm 2. Constrained Agglomerative Clustering.

Constrained Agglomerative Clustering (X, L, ML, CL)

\# ***X***: original data set; ***L:*** type of linkage {min, max, complete}

\# ***ML***: must links constraints; ***CL***: cannot link constraints

(1) Perform normalization on original data ***X*** to obtain the normalized data set ***R***

(2) Construct the transitive closure of the ***ML*** constraints resulting in r connected item sub-groups $\boldsymbol{R_{ML1}, \ldots, R_{MLr}}$

(3) Let $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_r$ be the cluster centroids of $\boldsymbol{R}_{ML1}, \ldots, \boldsymbol{R}_{MLr}$ to represent each sub-group
(4) If two points $\{\boldsymbol{x}, \boldsymbol{y}\}$ are both in ***ML*** and ***CL*** constraints $=>$ output "no solution" and stop
(5) Let $\boldsymbol{R}_1 = \boldsymbol{R} - \left(\cup_{i=1}^{r} \boldsymbol{R}_{MLi}\right)$. Let $\boldsymbol{k}_{max} = r + |R_1|$
(6) Construct initial feasible clustering $C_{\boldsymbol{k}_{\max}}$ with $k_{\max}$ clusters consisting of the r cluster $\boldsymbol{R}_{ML1}, \ldots, \boldsymbol{R}_{MLr}$ and a singleton cluster for each point in $\boldsymbol{R}_1$. Let $\boldsymbol{t} = \boldsymbol{k}_{max}$
(7) Construct an initial distance matrix D where pair-wise distance $\delta(c_i, c_j) = D_{ij}$, by initial feasible clustering $C_{\boldsymbol{k}_{\max}} = \{\boldsymbol{c_i}$ ***for each cluster i***$\}$
(8) $D =$ ***imposeCannotLinks (D, ML, CL)***
(9) While $\boldsymbol{t} > 1$ do
   (a) Select pair of closet $(\boldsymbol{c_m}, \boldsymbol{c_n}) = \boldsymbol{arg\ min}_{c_i, c_j \in C}\ \delta(\boldsymbol{c_i}, \boldsymbol{c_j})$
   (b) Merge c_m and c_n into c_{new} in C_{t-1} (***the result is Dendrogram***$_{\boldsymbol{t-1}}$)
   (c) ***updateDistanceMatrix (D, L,*** $\boldsymbol{C_{t-1}}, \boldsymbol{c_m}, \boldsymbol{c_n}, \boldsymbol{c_{new}}$)
   (d) $\boldsymbol{t} = {}_{\boldsymbol{t-1}}$
   End While
Return ***Dendrogram***

The pseudo code of the proposed COI-based constrained clustering method is shown in Algorithm 2. Essentially, after normalized the original item data set, the algorithm constructs the transitive closure of the ML constraints to generate $\boldsymbol{r}$ sub-groups of items $\boldsymbol{R}_{ML1}, \ldots, \boldsymbol{R}_{MLr}$. The computed centroids $m_1, \ldots, m_r$ of $\boldsymbol{r}$ sub-groups are used to replace the data points of sub-groups for clustering. The initial distance matrix is constructed by calculating pairwise distance among sub-groups by ML constraints and singleton data points which are not specified in ML constraints. In order to handle the CL constraints, the ***imposeCannotLinks*** function is called to reconstruct the distance matrix by resetting the distance between items to a relatively large numerical value if both of them are in CL constraints. The large distance in fact induces the clustering algorithm not to merge the items during clustering process. Besides, if any item in CL is also in ML with other items, the CL constraints will propagate to the items of the particular sub-group constructed by that ML. The distance of that propagated item should be reset accordingly.

The while loops in Algorithm 2 shows the merge process of agglomerative clustering. Basically, the items have the closest distance will be merged first. The distance matrix will be updated based on the specified linkage by calling ***updateDistanceMatrix*** function addressed in Algorithm 3. As defined in Equation (5), the new distance calculation which includes PC loadings of COI domain of each sub-group was proposed. Basically, if more than one item in the group, the PCs of each sub-group on retrieval frequency and required space can be constructed. The PC loadings are appended with centroids vector for distance calculation. It also means that during sub-group merging process, the PC loadings of COI for each sub-group will be compared. As well with centroids of sub-group, the smaller difference of PC loadings and centroids will induce the clustering algorithm to merge the sub-groups. If any item for distance calculation is singleton, the PC loadings cannot be calculated and there is no need to consider PCs loadings for singleton cluster:

Algorithm 3. updateDistanceMatrix.

updateDistanceMatrix (distance matrix ***D,*** type of linkage ***L,*** cluster set ***C,*** cluster $\boldsymbol{c_m}, \boldsymbol{c_n}, \boldsymbol{c_{new}}$)
   \# δ : pair-wise distance of two clusters by considering distance between the centroids of clusters

δ $(\boldsymbol{c_i}, \boldsymbol{c_j})$ = distance (centroids of $\boldsymbol{c_i}$, centroids of $\boldsymbol{c_j}$)
δ' : pair-wise distance between two non-singleton clusters. The centroids of clusters and first two PCs′ loadings on COI domain are considered for distance calculation
$\delta'(\boldsymbol{c_i}, \boldsymbol{c_j})$ = distance([centroids of $\boldsymbol{c_i}$ + $\boldsymbol{PC}$ loadings of $\boldsymbol{c_i}$], [centroids of $\boldsymbol{c_j}$+ $\boldsymbol{PC}$ loadings of $\boldsymbol{c_j}$])
    for $\boldsymbol{c_i} \in \boldsymbol{C}$ -$\boldsymbol{c_{new}}$
        if $|\boldsymbol{c_i}| > 1$ and $|\boldsymbol{c_m}| > 1$ and $|\boldsymbol{c_n}| > 1$
            δ_{new} $(\boldsymbol{c_i}, \boldsymbol{c_{new}}) = \boldsymbol{L}\ (\delta'(\boldsymbol{c_i}, \boldsymbol{c_m}), \delta'(\boldsymbol{c_i}, \boldsymbol{c_n}))$
        else
            $\delta_{\boldsymbol{new}}$ $(\boldsymbol{c_i}, \boldsymbol{c_{new}}) = \boldsymbol{L}\ (\delta(\boldsymbol{c_i}, \boldsymbol{c_m}), \delta(\boldsymbol{c_i}, \boldsymbol{c_n}))$
    return $\boldsymbol{D}$

Based on the constrained clustering method above, the items can be clustered by considering ML and CL constraints and data distribution on COI domain. The output dendrogram of clustering algorithm will be used for validating the clustering and determining the number of clusters. Due to the nature of agglomerative clustering, the dendrogram shows that all items merge together in the end (on the top of tree structure). The CL constraints in fact limit the number of clusters on certain range. Therefore, when determining the number of clustering, checking if CL constraints are violated under a particular number of clusters should be included in the clustering validation process.

Because the "true" number of clusters is usually unknown for warehouse data, the internal validation index which considers compactness and separation of clusters is commonly used to validate the clustering result and determine the number of clusters (Liu *et al.*, 2012). Several clustering validation indices such as the root mean squired error (RMSE), silhouette index, Davies-Bouldin, SD index, and so on can be used. In this research, without losing the generality, a validateCluster function whose pseudo code was shown in Algorithm 5 was proposed to validate the clustering result by specific validation index. Essentially, ***validateCluster*** will check the clustering label of any pairs of CL constraints. If they are clustered in the same group under k clusters, this clustering result is not valid and return null value. Otherwise, the specific validation index is computed and returned. By ***validateCluster***, the number of cluster k can be evaluated if CL constraints are violated under k clusters:

Algorithm 4. validateCluster.

validateCluster (dendrogram *tree,* validation index ***I,*** cannot Links ***CL,*** number of cluster ***k***)
tree : dendrogram of clustering result which can be cut by ***k*** clusters
α: A label vector contains clustering labels $\alpha_1,\ldots,\alpha_{\boldsymbol{n}}$ for each data point where $\alpha_{\boldsymbol{i}} \in \{1, .., \boldsymbol{k}\}$
    $\boldsymbol{L} = \boldsymbol{tree\ (k)}$
    $\alpha = \boldsymbol{Label\ (L)}$
    for $\boldsymbol{(i, j)} \in \boldsymbol{CL}$
        if $\alpha_{\boldsymbol{i}} == \alpha_{\boldsymbol{j}}$ # ***i*** and ***j*** are within the same cluster
            return ***NULL***
    return ***I (tree, k)***

3.4 Location assignment problem: COI-based assignment rule

The above clustering result divides all the items in the warehouse into multiple groups. Now the groups of items need to be assigned to warehouse layout. There are some characteristics should be concerned during location assignment such as amount of items, throughput (shipping frequency), storage space and material characteristics (weight, shape, size, stack ability, flammability, and environmental requirements, etc.). Besides, the storage method including storage equipment, devices or container for storing is also considered due to its effect to warehouse arrangement (Francis *et al.*, 1992). The heuristic assigning model based on COI rule first described by Heskett (1963) was applied to integrate with constrained clustering result. The COI-based assignment rule deliberates to list all items in ascending order of COI value and allocates the lowest of them to the most desirable location (Lai *et al.*, 2002).

The proposed assigning method contains two stages. The first stage is to assign the item clusters and it's sub-groups to a location or storage area which can meet the item cluster's constraints. Please note that the item cluster might contain multiple item sub-groups which are constructed by ML constraints addressed in the previous subsection, and singleton cluster which contains only one item. All clusters are first ranked in accordance with its number of storage constraints. Here, the constraints include the item-level ML and CL constraints and any storage restrictions such as container limit, and environmental constraints. For example, if the item cluster has the environmental requirement on drying area, only the storage areas which have drying environment can be selected. The cluster which has more constraints is processed first in order to reduce the difficulty of area searching. Then, the algorithm searches out all available storage areas which can meet each cluster's constraints. Within the available storage areas, the assignment continues to allocate the area closer to I/O point to the item sub-group with the lowest COI index. Since the proposed clustering algorithm has considered the PC of COI domain, the sub-groups with similar COI distribution will be clustered together in the cluster. This in fact can facilitate the assignment process to allocate the storage area which can particularly meet the need of a specific characteristic of COI distribution.

After determining the storage locations for a particular item cluster and its associated sub-group, the second stage of the algorithm is to assign the items in the cluster to the slot of that area based on the COI value of each item. The COI value of each item is calculated and then these values are sorted in the ascending order. Moreover, the distances from each storage lot to the I/O point are determined. Follow the COI-based assignment rule, the items with the lowest COI value will be allocated to the most desirable location which is closest to the I/O point (minimal distances). The same assignment procedure is applied on the singleton cluster which only contains one item. The notations used in the location assignment algorithm are listed below:

Notation	Description
i	Index of cluster
j	Index of I/O point
s	Index of item sub-group
u	Index of item
l	Index of storage location
Q	Total number of storage location in warehouse
NC	Number of clusters
L_i	The required space for cluster i
S_{is}	Number of sub-groups in cluster i
U_{isu}	Number of items in sub-group s in cluster i
U_{i0u}	Number of singleton items in cluster i

$Space_u$	Required space for item u
$Freq_u$	Retrieval frequency per period for item u
$Space_{\mathrm{is}}$	Required space for sub-group s in cluster i
d_{jl}	Ravel distance from I/O point j to storage slot l with m is the number of I/O points

The COI value of sub-group s in cluster i can be calculated by Equation (6), and COI value of item u are calculated as Equation (7):

$$COI_{sub-group\ is} = \frac{\sum_{u\ in\ sub-group\ is} Space_u}{\sum_{u\ in\ sub-group\ is} Freq_u} \tag{6}$$

$$COI_u = \frac{Space_u}{Freq_u} \tag{7}$$

The ***assignLocation*** function which addresses the pseudo code of assigning location for sub-group, item, and singleton cluster is listed in Algorithm 5. This function is called by the whole heuristic algorithm based on COI rule which is listed in Algorithm 6:

Algorithm 5. AssignLocation.

AssignLocation ($X \in$ {sub-group s, items in subgroup $\underline{u}$, singleton items u})
If X is a sub-group

$$COI_X = COI_{sub-group\ is}$$

Else

$$COI_X = COI_u$$

1. Sort in an increasing order of COI_X
2. Assign X with the lowest COI_X to the storage location l^* which is closest to I/O point ($min(d_{jl})$) until the required space for X ($Space_x$) is fulfilled
3. Mark the storage location l^* as an occupied location for X

Algorithm 6. assignLocationforClustering.

assignLocationforClustering(dendrogram *clusters*, ITEM_DATASET, *NC, Q*)
ITEM_DATASET contains raw data regarding items' characteristics
(1) Rank the clusters based on its constraints. The first rank is the cluster that has the most number of constraints.
(2) For $i = 1,\ldots,NC$ (start from assigning the first rank cluster)
  a. Search the storage location l' which can fulfill all constraints of cluster i
     $l' = []$
     For $l = 1,\ldots,Q$
        Verify if the specifications of storage area l can meet the constraints of cluster i. If yes, $l' = l' + l$
     End for
  b. Allocate location area to the cluster.
     If $|l'| \geqslant L_i$
        If $S_{is} > 0$
           For $s = 1, \ldots, S_{is}$

- ***assignLocation*** (sub-group (*is*))
- Assign items in sub-group *s*
  For $u = 1, \ldots, U_{isu}$
  - ***assignLocation*** (items in subgroup (*isu*))
  End for
End for
Else (consider the cluster contains the singleton item)
For $u = 1, \ldots, U_{i0u}$
- ***assignLocation*** (singleton items (*i0u*))
End for
End if
Else
Stop the algorithm because no storage area cannot be found for cluster *i*
End if
End for

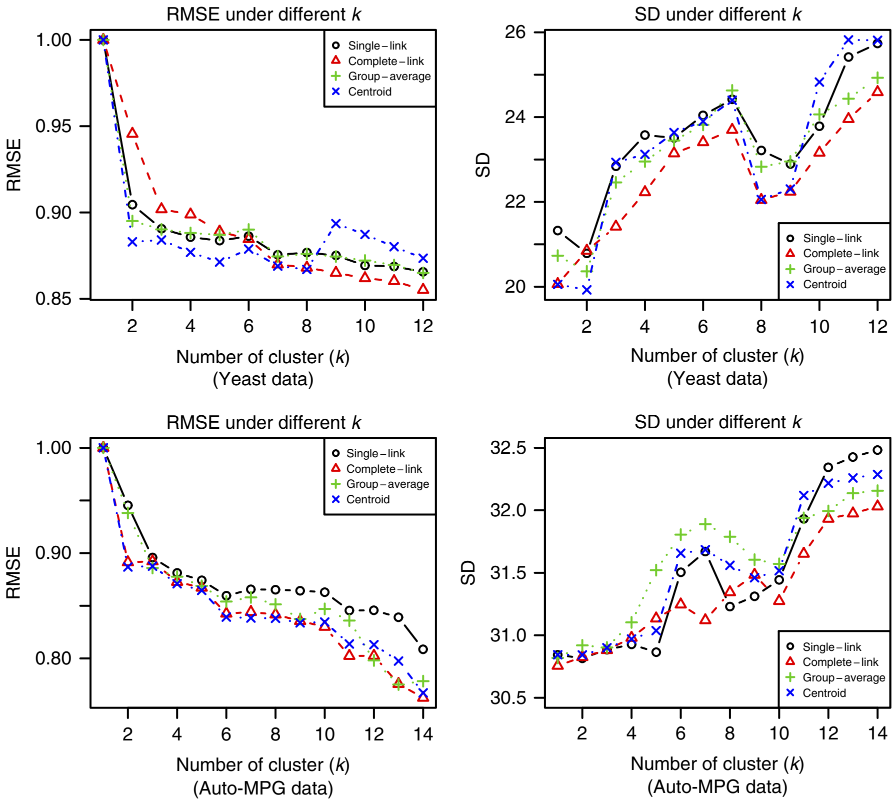
4. Results

In this section, two kinds of experiments were designed to evaluate the proposed method on UC Irvine Machine Learning Repository (UCI) data sets and warehouse data set mentioned in this case study, respectively. First, the public UCI datasets were used to compare the proposed method against agglomerative hierarchical methods with different linkages. Second, the real-world warehouse data set mentioned early was used to evaluate the clustering result and location assignment performance.

4.1 Experimental result on UCI data sets

Four major linkage methods including single-link, complete-link, centroid, and group-average are commonly used for computing the distance among clusters in traditional hierarchical clustering. In order to determine which linkage method should be applied in the proposed constrained clustering method, two datasets from UCI was used to compare the clustering results. The Yeast data set contains 1,484 instances with nine attributes including the class attribute which is related to the localization site of yeast protein. Also, the Auto-MPG data set concerns the city-cycle fuel consumption in miles per gallon including 398 data points which can be predicted by eight attributes. In this experiment, the attribute "class distribution" of Yeast data set, and "brand name" of vehicles in Auto-MPG data set were used to represent the ML constraints when the proposed method needs ML constraints for evaluation.

To evaluate the performance of the proposed method with four different linkages, two validation indices including RMSE and SD were used to compare the clustering results. The RMSE is commonly used in data clustering to evaluate the compactness of clusters. The SD indicator has two parts: SD-Scatter and SD-Dis. SD-Scatter indicates the average compactness of clusters while SD-Dis specifies the degree of separation between the clusters. Figure 3 shows the comparison results of linkage methods which were applied in the proposed method. As can been seen, the complete-link method tends to produce more compact clusters where the RMSE values are smaller in the two datasets, especially when number of clusters increases. Similarly, the SD values of the complete-link are also smaller than the others. Therefore, the complete-link method is selected to use in the proposed constrained clustering method for computing the distance among data points.



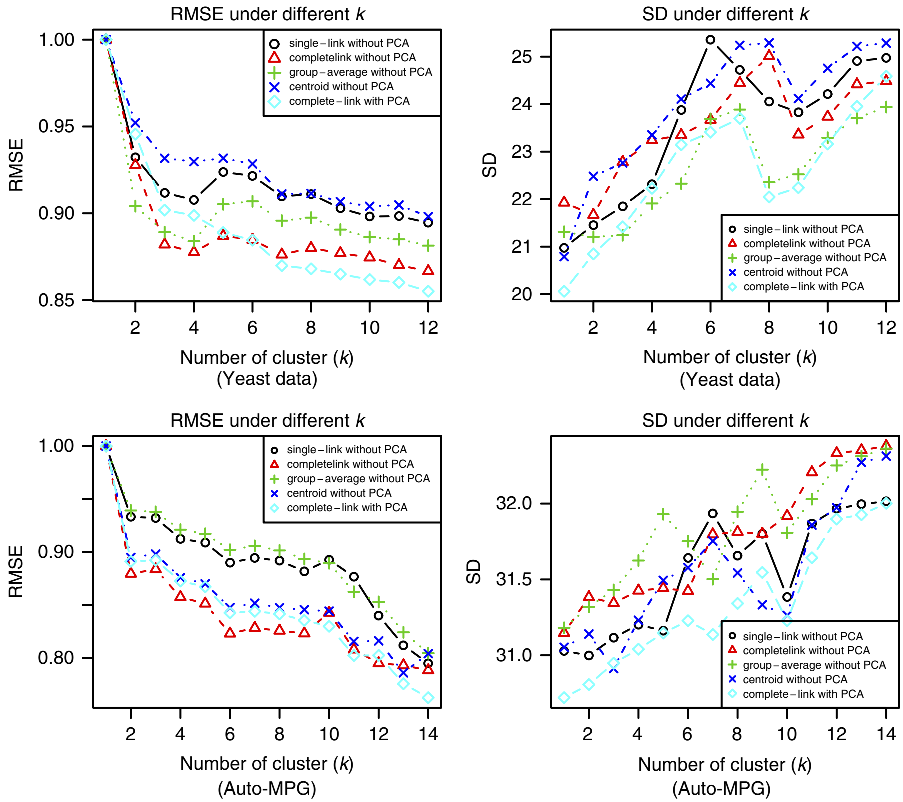
Note: RMSE and SD indices were evaluated on UCI Yeast and Auto-MPG data set

Figure 3.
The comparison of linkage methods applied in the proposed method

Next, the proposed constrained clustering with complete-link and PCA method was compared with traditional agglomerative hierarchical clustering methods. Similar to the experiment above, four different linkage methods in traditional hierarchical clustering without applying PCA were the benchmarks in this experiment. Figure 4 shows the comparison results of RMSE and SD on the UCI Yeast and Auto-MPG data set with the same ML constraints settings. As can been seen in SD results, the proposed constrained hierarchical clustering (complete-link with PCA) has lower SD values which means that it outperforms other linkage methods based on both Yeast and Auto-MPG data set. Considering the RMSE result, it seems the proposed method (complete with PCA) only dominates single-link, group-average and centroid linkage. Although the proposed method has similar performance with complete-without-PCA in term of RMSE values, by consider RMSE and SD combined, we still can conclude that the proposed method (complete with PCA) has better performance in general.

4.2 Experimental result for warehouse data

In order to evaluate the proposed clustering method and the heuristics location assignment method based on warehouse data in HY, two steps of evaluations were performed. The first step evaluated the clustering results with and without considering CL constraints and PCA under the same group-level ML constraints. Because the



Note: RMSE and SD indices were evaluated on UCI Yeast and Auto-MPG data set

Figure 4. The comparison of the proposed constrained clustering method against traditional agglomerative hierarchical clustering

existing warehouse only utilized ML constraints to group items for dedicate storage and the human intervention was conducted to determine the location assignment, the existing item clustering result was the benchmark for the comparison. The second step of evaluation focusses on simulating the retrieval distance under dedicate and class-based storages. The total retrieval distances of one-year operation was used as the performance measure to compare the efficiencies of dedicate and class-based storage.

4.2.1 Comparison of clustering results. The proposed constrained agglomerative clustering was used to perform the clustering analysis on the data set of the stored items which contains information about volume; container size; retrieval frequency; producing quantity per kilogram; and batch size of shipment of each item. Totally, 435 items were pre-determined to form the sub-groups based on ML constraints of their material type and secondary process in the mentioned warehouse.

Figure 5(a) shows the clustering result without considering the CL constraints and PCA distribution patterns among item ML groups. The dendrogram of clustering shows that, without considering the CL constraints, the clustering method clusters group 3 and 7 together in early agglomerative process which violates the CL constraints mentioned beforehand. In fact, this result constrains the number of cluster k to be at least 12 (see horizontal dashed line) for not to violate CL. This relatively large k will make the clustering on 14 sub-groups less usage.

Figure 5(b) shows the clustering results after considering CL constraints and PCA of item sub-groups. In general, four major portions of item sub-groups can be recognized in Figure 5(b) which is quite different from Figure 5(a). Especially, groups 3, 13, and 12 which were clustered as an independent cluster are not merged with other groups in the end. Also, by considering the CL constraints simultaneously with PCA information, the group 3 is clustered apart from group 7. This clustering result is consistent with the expectation when the CL constraints were formed to restrict group 3 to be together with group 7.

As mentioned above, RMSE and SD index are used as the two validation indices to compare the clustering result. In Figure 6, RMSE and SD of the clustering results under different k are plotted. Both result shows that the compactness of the constrained clustering method which considers the PCA patterns and CL constraints is relatively smaller than it without considering CL and PCA patterns.

The RMSE and SD plot were reviewed to determine the number of clusters. The optimal number of clusters is conventionally selected as the elbow point in the RMSE plot. For SD index, the number of clusters with minimum SD is commonly used. Based on the clustering result, we can see the RMSE drops dramatically when number of clusters reaches 2 and continue to reduce until $k = 8$. However, in SD plot, the minimum SD is at $k = 3$ and the minimum SD of the second lowest valley is at $k = 8$. Therefore,

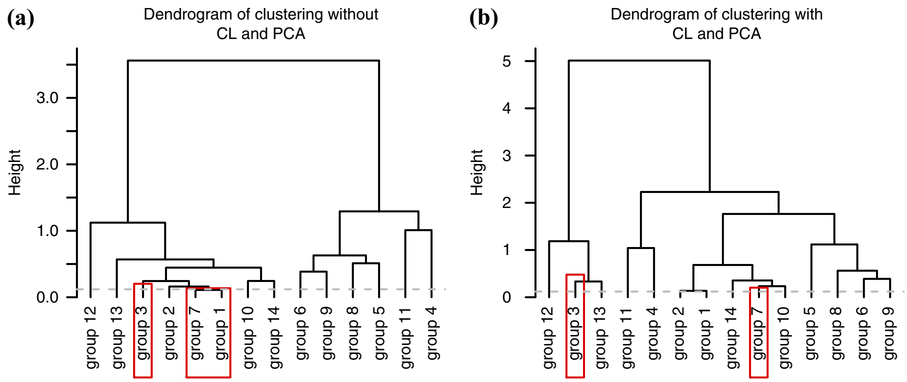


Figure 5.
The clustering result without and with CL constraints and PCA loadings on warehouse data set

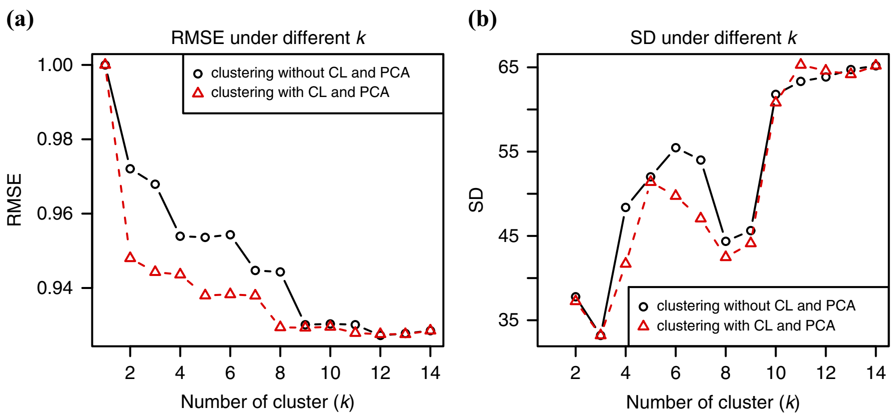


Figure 6.
The comparison of RMSE and SD between clustering result without and with CL constraints and PCA loadings on warehouse data set

combined the suggestions of both indices, eight-cluster tends to be a cut-off point for determining the number of clusters in this case.

Figure 7 shows the plot of required space against retrieval frequency (COI domain) of eight clusters. As can be seen, after performing the proposed clustering method, the sub-groups in each cluster have similar PC vectors. Therefore, the difference on COI distribution between clusters is obvious. It means the proposed algorithm is able to differentiate the COI distribution by using PC loadings. For example, cluster (2) containing groups 3 and 13 and cluster (6) containing groups 1 and 2 have very closed centroid points but different COI distributions. This separation demonstrates that the clustering not only consider the centroids of the sub-group but also the COI distributions for better clustering.

4.2.2 Comparison of location assignments. Based on the clustering result, the location assignment algorithm introduced beforehand was used to assign the grouped items in the warehouse. The characteristics of items were considered when assigning the items on the shelves in the different areas. In this case study, because the basic characteristics of items and intra-constraints have been clustered by the proposed constrained clustering method in each sub-group, the location assignment problem can be simplified to assign the item clusters to the storage area first, then assign item in each clusters.

Based on the developed assigning algorithm, the item clusters were ranked by the number of constraints which should be fulfilled. Then, all storage areas are verified to search the available area for a particular cluster. Once the storage area is identified, the COI was computed for each sub-group and each item in the cluster. The shelf closer to shipping and packing zone is assigned for the item which has smaller ratio.

In order to evaluate the new location assignment, the total retrieval distance of all items under the existing warehouse and new location assignment with ML, CL, and PCA were computed based on one-year retrieval history. Please note that the new location assignment was conducted by clustering result of the proposed clustering method. The storage efficiency was calculated by the product of the distance from the storage area to shipping area multiply the retrieval times. Based on this calculation, the total storage retrieval distance of the new assignment is 43,380.5 meter which is much smaller than the one by the existing assignment 65,238 meter. Roughly speaking, around 33 percent of retrieval distance can be saved by the new location assignment in this case study. This result also convicted the management echelon of the better efficiency by class-based storage. The developed constrained clustering method has been implemented in HY’s warehouse system to study the location assignment problem.

5. Conclusions

5.1 Summary of findings

This case study paper addressed a practical location assignment problem which concerns the storage constraints among items for a class-based warehouse. In order to answer the research gap about how to cluster items for class-based storage, this paper first addressed constrained clustering method to construct ML and CL constraints which can be formed by practical storage operating. The proposed data clustering framework identified three major steps to handle ML constraints based on storage requirement; CL constraints among items and item ML groups; and PCA among pre-determined item sub-groups to better differentiate item clusters by considering COI distribution. The proposed clustering algorithm can cluster the items to fulfill the ML and CL constraints and also differentiate the COI distribution of item sub-groups. Then the clustering result was applied in the heuristics algorithm to arrange location in

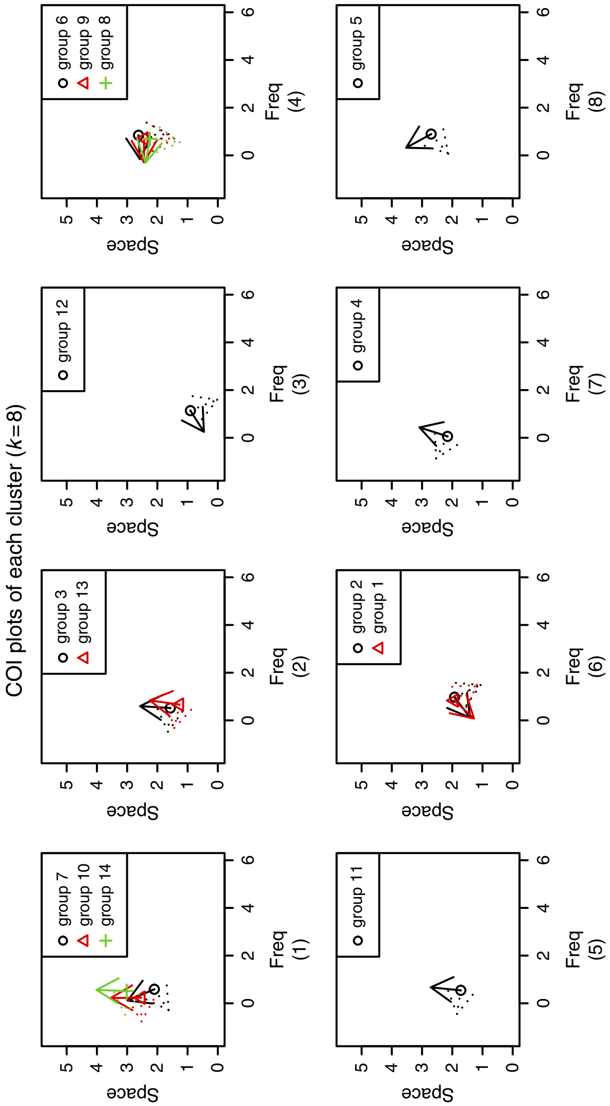


Figure 7.
COI plots of each cluster under $k=8$ by the proposed clustering method. For each cluster, multiple sub-groups might exist

warehouse. Based on the physical or functional limit of the warehouse, the proposed assigning algorithm can assign and locate the group of items which are similar in terms of either item characteristic or storage requirement to the warehouse.

A real-world case study was addressed in this paper to demonstrate an example of practical location assignment problem. Based on the data clustering result, the developed constrained clustering method which considering CL constraints and PCA outperforms the traditional hierarchical clustering method in terms of RMSE and SD. It shows that the PCA analysis is able to contribute on selecting more compact clusters. Considering CL constraints not only fulfill the storage requirement among items but also identify the intra-clustering to simplify the location assignment. The result shows that the developed algorithm is able to cluster the stocks based on their criteria and provide the better compactness within the clusters. Compared with the location assignment before and after applying the developed framework, the simulation result shows the retrieval efficiency can be improved by 33 percent in terms of shortening the travel distance.

5.2 Implications and future research

Although the case study might only represent certain warehouse operation with specific storage constraints, this paper shreds a light of developing a data clustering method which can be directly applied on solving the practical data analysis issues. While number of items in warehouse is tremendously large such as Amazon's regional warehouse, the complication among item and location constraints will lead to the complex item clustering and assignment problem. The developed method can be easily fit in to solve the item clustering and assignment problem. By utilizing the proposed constrained clustering method, the item-item and item-location constraints can be converted to multiple ML and CL constraints first. Then, the agglomerative hierarchical clustering with PCA method can help on revealing the better clustering results and also analyzed the COI as the item assignment references. Based on the clustering results, the developed heuristic item assignment which considers traveling distance and storage constraints at the same time, will ensure the better efficiency of item retrieval and enhance the warehouse management.

In the future work, the constrained K-mean method can be studied to compare with the proposed method. For improving the solution quality of location assignment problem, the meta-heuristics method such as genetic algorithm or practical swarm optimization integrating with constrained clustering method can be developed to search for better solution. The multi-objective methodology can be used to optimize not only the retrieval distance but also space utilization for each item class. In addition, the work can be extended to handle the vertical adjacent constraints which provide the detailed allocation within multiple-layer storage rack.

References

Davidson, I. and Ravi, S.S. (2005a), "Agglomerative hierarchical clustering with constraints: theoretical and empirical results", in Jorge, A., Torgo, L., Brazdil, P., Camacho, R. and Gama, J. (Eds), *Knowledge Discovery in Databases: PKDD 2005*, Springer, Berlin and Heidelberg, pp. 59-70.

Davidson, I. and Ravi, S.S. (2005b), "Clustering with constraints: feasibility issues and the k-means algorithm", in Kargupta, H., Srivastava, J., Kamath, C. and Goodman, A. (Eds), *Fifth SIAM International Conference on Data Mining*, SIAM, Newport, CA, pp. 138-149.

Egas, C. and Masel, D. (2010), "Determining warehouse storage location assignments using clustering analysis", International Material Handling Research Colloquium, Milwaukee.

Eynan, A. and Rosenblatt, M.J. (1994), "Establishing zones in single-command class-based rectangular AS/RS", *IIE Transactions*, Vol. 26 No. 1, pp. 38-46.

Francis, R.L., McGinnis, L.F. and White, J.A. (1992), *Facility Layout and Location: An Analytical Approach*, Prentice Hall, Englewood Cliffs, NJ.

Frazelle, E. (2001), *Supply Chain Strategy*, McGraw-Hill Trade, New York, NY.

Hausman, W.H., Schwarz, L.B. and Graves, S.C. (1976), "Optimal storage assignment in automatic warehousing systems", *Management Science*, Vol. 22 No. 6, pp. 629-638.

Heskett, J.L. (1963), "Cube-per-order index-a key to warehouse stock location", *Transportation and distribution Management*, Vol. 3 No. 1, pp. 27-31.

Jane, C.-C. and Laih, Y.-W. (2005), "A clustering algorithm for item assignment in a synchronized zone order picking system", *European Journal of Operational Research*, Vol. 166 No. 2, pp. 489-496.

Jiang, H., Ren, Z., Xuan, J. and Wu, X. (2013), "Extracting elite pairwise constraints for clustering", *Neurocomputing*, Vol. 99, pp. 124-133.

Jlloffe, I. (2005), "Principal component analysis", *Encyclopedia of Statistics in Behavioral Science*, Wiley, New York, NY.

Johnson, R.A. and Wichern, D.W. (2002), *Applied Multivariate Statistical Analysis*, Prentice Hall, Upper Saddle River, NJ.

Kouvelis, P. and Papanicolaou, V. (1995), "Expected travel time and optimal boundary formulas for a two-class-based automated storage/retrieval system", *International Journal of Production Research*, Vol. 33 No. 10, pp. 2889-2905.

Lai, K.K., Xue, J. and Zhang, G. (2002), "Layout design for a paper reel warehouse: a two-stage heuristic approach", *International Journal of Production Economics*, Vol. 75 No. 3, pp. 231-243.

Lambert, D.M., Stock, J.R. and Ellram, L.M. (1998), *Fundamentals of Logistics Management*, Irwin/McGraw-Hill.

Larson, T.N., March, H. and Kusiak, A. (1997), "A heuristic approach to warehouse layout with class-based storage", *IIE Transactions*, Vol. 29 No. 4, pp. 337-348.

Liu, C.-M. (1999), "Clustering techniques for stock location and order-picking in a distribution center", *Computers & Operations Research*, Vol. 26 Nos 10/11, pp. 989-1002.

Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J. and Wu, S. (2012), "Understanding and enhancement of internal clustering validation measures", *IEEE Transaction on Cybernetics*, Vol. 43 No. 3, pp. 982-994.

MacQueen, J.B. (1967), "Some methods for classification and analysis of multivariate observations", in Cam, M.L. and Neyman, J. (Eds), *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, CA, pp. 281-297.

Mantel, R.J., Schuur, P.C. and Heragu, S.S. (2007), "Order oriented slotting: a new assignment strategy for warehouses", *European Journal of Industrial Engineering*, Vol. 1 No. 3, pp. 301-316.

Muppani, V.R. and Adil, G.K. (2008a), "A branch and bound algorithm for class based storage location assignment", *European Journal of Operational Research*, Vol. 189 No. 2, pp. 492-507.

Muppani, V.R. and Adil, G.K. (2008b), "Class-based storage-location assignment to minimise pick travel distance", *International Journal of Logistics Research and Applications*, Vol. 11 No. 4, pp. 247-265.

Muppani, V.R. and Adil, G.K. (2008c), "Efficient formation of storage classes for warehouse storage location assignment: a simulated annealing approach", *Omega*, Vol. 36 No. 4, pp. 609-618.

Petersen, C.G., Aase, G.R. and Heiser, D.R. (2004), "Improving order-picking performance through the implementation of class-based storage", *International Journal of Physical Distribution & Logistics Management*, Vol. 34 No. 7, pp. 534-544.

Rosenwein, M.B. (1994), "An application of cluster analysis to the problem of locating items within a warehouse", *IIE Transactions*, Vol. 26 No. 1, pp. 101-103.

Shaohong, Z. and Hau-San, W. (2009), "Active constrained clustering with multiple cluster representatives", *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, *San Antonio, TX, October 11-14*, pp. 1103-1110.

Smith, L.I. (2002), "A tutorial on principal components analysis", available at: www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf (accessed April 7, 2016).

Sooksaksun, N., Kachitvichyanukul, V. and Gong, D.-C. (2012), "A class-based storage warehouse design using a particle swarm optimisation algorithm", *International Journal of Operational Research*, Vol. 13 No. 2, pp. 219-237.

Van den Berg, J. (1996), "Class-based storage allocation in a single-command warehouse with space requirement constraints", *International Journal of Industrial Engineering*, Vol. 3 No. 1, pp. 21-28.

Van Den Berg, J.P. (1999), "A literature survey on planning and control of warehousing systems", *IIE Transactions*, Vol. 31 No. 8, pp. 751-762.

Vishkaei, B.M. and Rashti, M.E.M. (2011), "A fuzzy multi-objective class based storage location assignment", *International Journal of Applied*, Vol. 1 No. 1, pp. 19-33.

Wagstaff, K. and Cardie, C. (2000), "Clustering with instance-level constraints", in Langley, P. (Ed.), *Proceedings of the Seventeenth International Conference on Machine Learning (ICML '00)*, Morgan Kaufmann Publishers Inc., San Francisco, CA, pp. 1103-1110.

Wagstaff, K., Cardie, C., Rogers, S. and Schr, S. (2001), "Constrained K-means clustering with background knowledge", in Brodley, C.E. and Danyluk, A.P. (Eds), *Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01)*, Morgan Kaufmann Publishers Inc., San Francisco, CA, pp. 577-584.

Zhao, W., He, Q., Ma, H. and Shi, Z. (2012), "Effective semi-supervised document clustering via active learning with instance-level constraints", *Knowledge and Information Systems*, Vol. 30 No. 3, pp. 569-587.

About the authors

Chao-Lung Yang received the BS Degree in Mechanical Engineering and MS Degree in Automation Control from the National Taiwan University of Science and Technology, Taiwan, in 1996 and 1998, respectively. He also received the MSIE and PhD Degree in Industrial Engineering from the Purdue University, West Lafayette, in 2004 and 2009, respectively. He is currently with Department of Industrial Management, National Taiwan University of Science and Technology, Taiwan, as an Assistant Professor. His research area is data mining, big data analytics, scheduling, and metaheuristic algorithm. He is a Member of IEEE and INFORMS. Chao-Lung Yang is the corresponding author and can be contacted at: clyang@mail.ntust.edu.tw

Thi Phuong Quyen Nguyen received the BS Degree in Industrial Systems Engineering from the Hochiminh City University of Technology in 2008 and MS Degree in Industrial Management from the National Taiwan University of Science and Technology, Taiwan, in 2013. She is currently continuing her PhD Degree in the Department of Industrial Management, National Taiwan University of Science and Technology. Her research interests include data mining, assembly line balancing, and meta-heuristic approaches.