Data mining based algorithm for storage location assignment in a randomized warehouse

King-Wah Pang*

Department of Logistics and Maritime Studies,

The Hong Kong Polytechnic University, Hung Hom, Hong Kong

Hau-Ling Chan

The Institute of Textiles and Clothing,

The Hong Kong Polytechnic University, Hung Hom, Hong Kong

September 2016

Abstract

Data mining has long been applied in information extraction for a wide range of applications such as customer relationship management in marketing. In the retailing industry, this technique is used to extract the consumers buying behavior when customers frequently purchase similar products together; in warehousing, it is also beneficial to store these correlated products nearby so as to reduce the order-picking operating time and cost. In this paper, we present a data mining based algorithm for storage location assignment of piece-picking items in a randomized picker-to-parts warehouse by extracting and analysing the association relationships between different products in customer orders. The algorithm aims at minimizing the total travel distances for both put-away and order-picking operations. Extensive computational experiments based on synthetic data that simulates the operations of a computer and networking products spare-parts warehouse in Hong Kong have been conducted to test the effectiveness and applicability of the proposed algorithm. Results show that our proposed algorithm is more efficient than the closest open location and purely dedicated storage allocation systems in minimizing the total travel distances. The proposed storage allocation algorithm is further evaluated with experiments simulating larger scale warehouse operations. Similar results on the performance comparison among the three storage approaches are observed. It supports the proposed storage allocation algorithm is applicable to improve the warehousing operation efficiency if items have strong association among each other.

Keywords

Data mining; association rules; warehousing operations; storage location assignment problem; order-picking; put-away.

*Corresponding Author. Tel.: +852 2766-4126; fax: +852 2330-2704

Email address: anthony.pang@polyu.edu.hk;

Data mining based algorithm for storage location assignment in a randomized warehouse

Abstract
Data mining has long been applied in information extraction for a wide range of applications such as customer relationship management in marketing. In the retailing industry, this technique is used to extract the consumers buying behavior when customers frequently purchase similar products together; in warehousing, it is also beneficial to store these correlated products nearby so as to reduce the order-picking operating time and cost. In this paper, we present a data mining based algorithm for storage location assignment of piece-picking items in a randomized picker-to-parts warehouse by extracting and analysing the association relationships between different products in customer orders. The algorithm aims at minimizing the total travel distances for both put-away and order-picking operations. Extensive computational experiments based on synthetic data that simulates the operations of a computer and networking products spare-parts warehouse in Hong Kong have been conducted to test the effectiveness and applicability of the proposed algorithm. Results show that our proposed algorithm is more efficient than the closest open location and purely dedicated storage allocation systems in minimizing the total travel distances. The proposed storage allocation algorithm is further evaluated with experiments simulating larger scale warehouse operations. Similar results on the performance comparison among the three storage approaches are observed. It supports the proposed storage allocation algorithm is applicable to improve the warehousing operation efficiency if items have strong association among each other.

Keywords
Data mining; association rules; warehousing operations; storage location assignment problem; order-picking; put-away.

1. Introduction

Efficient warehousing management is of vital importance for supply chain operations. This is due to the fact that warehouse acts as an intermediary connecting upstream suppliers with downstream customers along the supply chain. In order to enhance the competiveness, many companies attempt to achieve high volume production and distribution while keeping minimal inventories throughout the supply chain where products are to be delivered to customers within a short time (Berg and Zijm, 1999). Hence, efficient warehousing operations significantly reduce the order-picking distance and processing time of product movement for order fulfilment inside a warehouse so as to respond to customer requests faster with lower cost.

In warehousing management, operations can mainly be categorized into six activities: as receiving, transfer and put-away, order-picking/selection, accumulation/sortation, cross docking, and shipping. Among these activities, order-picking is the most labor-intensive operation under manual systems and capital-intensive operation under automated systems. Some studies suggest that order-picking costs can contribute to more than 55% of the total warehousing operation cost (Tompkins *et al*., 2003), and hence, order-picking operations should be optimized. Meanwhile, assigning the Stock Keeping Units (SKUs) to the right storage location has a significant impact on the order-picking operation in terms of travel distance required to retrieve the requested SKUs. Roodbergen and Koster (2001) have presented four approaches to reduce travel time or distance for order-picking activity: (1) determining good order-picking route; (2) zoning the warehouse; (3) assigning products to the right storage locations; (4) picking orders in batches. In order to achieve operations efficiency on warehousing, which is the composition of put-away and order-picking operations, decision on storage location assignment should be well planned prior to the order-picking operations.

In the literature on storage location assignment problems, there are several ways of assigning products to storage locations and the most frequently used location assignment policies are: closest open location, dedicated storage, and class based storage. Closest open location policy assigns the items at the first empty location closest to the I/O entry point. This policy reduces the put-away travel distance in most cases.

In dedicated storage policy, each product is assigned to a predetermined location. All locations are reserved for each product even when that product is out of stock. This approach has the advantage that order pickers become familiar with product locations, so they take shorter time to locate the products.

Closest open location and dedicated storage are two extremes of storage location assignment policy and they have been compared for their best use in literature. Existing studies have demonstrated that closest open location policy tends to give a better result in terms of order-picking travel distance when the warehouse utilization rate is low; while a dedicated turnover rate based storage policy is better when the warehouse space utilization is high (Linn and Wysk, 1987; Malmborg, 1996; Gu, 2005).

For automatic storage and retrieval system (AS/RS), when the closest open location policy for storage allocation and shortest-leg policy for crane movement are adopted, with the location to product ratio (LTPR) > 1, i.e., there are more locations than number of products stored in the warehouse such that multiple items of the same products may be stored, the closest open location strategy outperforms other storage schemes such as the full turnover dedicated policy in terms of total crane travel time (Gagliardi *et al.* 2012). Meneghetti and Monti (2014) proposed a dynamic energy-based strategy which is similar to the closest open location in time-based perspective. The result shows that the proposed dynamic storage policy outperforms the full turnover-based strategy when energy consumption is considered. It affirms that the closest open location policy is an effective storage policy when the location to product ratio is larger than 1.

For the class based storage policy, the storage locations are grouped into different classes, which based on the item popularity. Within each class zone, the items are normally stored nearest to the depot within the locations of that class. If the number of classes equals to the number of items, it is equivalent to the dedicated storage; If the number of classes equals to one, then it is closest open location policy. Otherwise, it is called the class based storage. In order to determine the class grouping, the importance of the items is considered which based on ABC Classification scheme according to Pareto's Law, or the 80-20 rule, which states that 80% of the demand is satisfied with only 20% of the stock-keeping units. Companies classify items into Class A, B and C according to their popularity and contribution to the total company sales volume. The purpose of determining the popularity is to identify the most frequently ordered items and allocate them to strategic locations so that the distance traveled for both put-away and picking of these items is reduced.

Glock and Grosse (2012) have presented a U-shaped shelving systems for the comparison of different storage policies and order picking strategies. Recently, Subir & Gajendra (2013) have proposed a 3-class storage policy for a S-shaped traversal routing system in a low-level picker-to-part warehouse. Quintanilla *et al.* (2015) have studied a unit-load warehouse storage location assignment problem with dynamic storage and practical restrictions, i.e., heavy, fragile or dangerous materials. They have introduced several heuristic algorithms to determine the strategies for relocating the stored items in order to improve the space availability. Guo *et al.* (2015) have studied the impact of the storage space requirement for zoning on different storage policies for a unit-load warehouse. Battini *et al.* (2015) have proposed a combined method for the order picking system design that considers both storage assignment and travel distance estimation. However, these storage policies do not consider interdependence among the products in a picking order.

In the retailing industry, we realize that customers frequently purchase similar products together; in warehousing operation, we observe a similar scenario that several items are frequently requested in the same picking order and these products are called correlated products. It is beneficial to store these items nearby to reduce travel time during the order-picking operation if the product dependency can be predicted. Correlated storage policy is based on the estimation of an appropriate index of correlation among the items in a product mix. In particular, cluster analysis identifies groups of items that customers frequently order together, products with high values of correlation are then stored near to each other as it reduces the travel distance of picking an order.

The strategy for the storage location assignment of the correlated products has also been studied extensively over the last two decades. Several products are frequently requested in one order and that these correlated products are suggested to be stored nearby to reduce travel time and orientation time during order-picking (Frazelle and Sharp, 1989). Frazelle (1990) has further studied and formulated the storage location assignment problem (SLAP) as an NP-hard mathematical programme and proposed a two-phrase construction heuristic for solving it. Lee (1992) has presented a storage assignment procedure to identify items with high correlation and suggested to assign these items to be stored closely in the storage rack according to the space-filling curve. Kim (1993) has also presented a correlated storage assignment procedure with the consideration of inventory related costs and material handling costs for both discrete-picking and batch-picking operations. Liu (1999) has applied the clustering techniques to extract the correlation information of the customer orders, which is subsequently used for optimizing the stock location and picking process. Bindi *et al*. (2008) have compared different storage allocation rules in a warehouse with correlated storage policy and proposed a heuristic algorithm based on the similarity index among the products for storage location assignment.

Chiang *et al*. (2008 & 2011) have presented a heuristic storage assignment method based on association rule mining technique to reduce the travel distance of order-picking operation. Xiao and Zheng (2009) have studied the correlated storage location assignment in a production warehouse with limited and deterministic product bill of material information. Chan *et al.* (2010) have also presented the concept of using data mining technique for the storage location decision in a randomized warehouse. Chuang et al. (2012) have proposed a clustering strategy based on the association between items so that they are stored in the same cluster for more efficient order picking operations. Recently, Li et al. (2015) developed a genetic algorithm to solve the dynamic storage assignment problems with the consideration of the product pair-wise association and ABC classification. The result of their studies shows that this direction of research is promising and worthy of study in more detail. However, those previous studies presented do not consider the minimization of both the put-away and order picking distance, most studies consider the minimization of the order picking operations only, except the work Chan et al. (2010).

In recent years, the development of information technology is growing rapidly in the applications of supply chain management. Enterprise resource planning (ERP) system, radio frequency identification (RFID) system, and warehousing management system (WMS) are widely used in companies for improving the operation efficiency. Besides, recording and gathering transaction data of customer orders is more timely and cost efficient than before. Knowing the customer order patterns from transaction history is crucial not only for increasing the sales volume in the retail industry, but also for the facilitation of better decision making in warehousing management. Through capturing the order transactions by data mining technique, companies can analyse the interdependence of the product selling pattern easily and effectively. One of the most popular applications of data mining in warehousing is the Market Basket Analysis by association rules extraction which extracts the information of the set of products purchased together by a customer in a single store visit (Chen *et al*., 2005).

Association rules mining can be described as follows: Given a set of items defined as $\mathrm{I} = \{\mathrm{I_i}, \mathrm{i} = 1, 2, \ldots, \mathrm{m}\}$. Let D be a set of transactions, where each transaction T is a set of items such that $\mathrm{T} \subseteq \mathrm{I}$. We say that a transaction T contains X, a set of some items in I, if $\mathrm{X} \subseteq \mathrm{T}$. An association rule is an implication in the form of $\mathrm{X} \Rightarrow \mathrm{Y}$, where $\mathrm{X} \subseteq \mathrm{I}$, $\mathrm{Y} \subseteq \mathrm{I}$, and $\mathrm{X} \cap \mathrm{Y} = \emptyset$. Such rule implies that if customers buy itemset X, they will also buy itemset Y simultaneously.

The mining of association rules is to generate all association rules that have support and confidence level greater than the user-specified thresholds. The support of an association rule $X \Rightarrow Y$ is the percentage of transactions in the database that contains $X \cup Y$ which means the rule has support s in the transaction set D if s% of transactions in D contains $X \cup Y$. For the confidence of an association rule $X \Rightarrow Y$, it is the ratio of the number of transactions that contains $X \cup Y$ to the number of transactions that contains X . That means the rule $X \Rightarrow Y$ holds in the transaction set D with confidence c, if c% of transactions in D that contains X also contains Y. Both support and confidence are jointly taken as measures of association between any set of items and reflect how often the association occurs and how strong is the rule in the dataset respectively (Agrawal *et al*., 1993).

Apart from the storage policies for the put-away operations, order picking problems have also been studied extensively in the literature. Some researchers have studied the formulation of the order-picking operation as an NP-hard problem and developed heuristic algorithms that minimize the total distance traveled by the order pickers or storage/retrieval machines (Elsayed, 1981; Elsayed and Stern, 1983; Hwang *et al*., 1988; Hwang and Lee, 1988; Elsayed and Unal, 1989; Gibson and Sharp, 1992; Rosenwein, 1996).

Most of the existing literature study the order-picking operation without considering the storage location assignment when replenishing the SKUs. Therefore, in this paper, we propose an algorithm that primarily focuses on discrete order-picking operations and applies data mining technique to identify the correlated products and determine the storage locations and put-away operations of these items in a randomized picker-to-parts warehouse. The objective of this study is to develop a storage location assignment algorithm that minimizes the manual efforts in the warehousing operations by optimizing the total distance traveled on both put-away and order-picking operations.

The rest of the paper is organized as follows: In Section 2, we define the storage location assignment problem and the optimization objective. In Section 3, the formulation and the proposed methodology are presented. Section 4 describes the design of the experimental study and the results used for comparing the performance between our proposed method and two other storage location assignment approaches. Finally, the paper is summarized with some concluding remarks and future research direction in Section 5.

2. Problem description

Layout and operational parameters play a significant role in the storage location assignment decision (Bindi *et al.*, 2008) as it impacts the total travel distance generated and the time associated with the storage and retrieval activities. In this paper, we consider a picker-to-parts warehouse of unit-load items. The layout of the warehouse for the experimental study is presented in Figure 1. The warehouse has M vertical storage shelves. Each shelf has M discrete pick locations of uniform size on both sides of the shelf. Each pick location can store only one unit of an item. There is a single I/O point located at the lowest-left hand corner of the warehouse. There are end-aisles available on both ends of the shelves for the order picker to traverse from one shelf to another shelf. There is no cross-aisle available between the shelves. The put-away and order picking operations are performed manually using an order-picker truck to reach the storage location for put-away and order picking operations. The warehousing operation adopts a multi-command policy that put-away or order picking operation of multiple items is performed in one trip.

The objective of the proposed storage location assignment algorithm is to determine the strategic location for storing the replenish products such that correlated products are placed closely in order to minimize the total travel distance for both put-away and order-picking operations. The association among the products and the popularity of each product can be extracted and determined by analysing the order history using data mining algorithm.

In a picker-to-parts warehouse, the order-picking and put-away operations are executed as follows: when a customer order is received in a warehouse for order-picking, the requested items and the corresponding quantities are listed on an order pick list. The order picker reaches the storage location and collects the piece-picking items from the shelf according to the pick list following the system-generated pick route. Then, the pick truck returns to the I/O point for packing and shipment preparation. When the stock level of an item in the warehouse is not enough to satisfy the future customer demand or reach a predetermined level for replenishment, a replenishment order is placed to refill the item. Once the order of the item is received, the item is being put-away to the storage location based on the storage location decision system.

We assume to have complete information regarding the customer orders for picking and the replenishment orders for put-away operation.

To formulate the storage location assignment problem of the put-away operation as a mathematical programming model, we first define a set of $\mathbf{P}$ items to be put-away to the warehouse, with $|\mathbf{P}| = p$. Besides, we have a set of empty locations available in the warehouse for storing items, defined as $\mathbf{L} = \{1, 2, \ldots, |\mathbf{L}|\}$. We assume $|\mathbf{L}| \geq p$ that there are enough empty spaces to store the arriving items. The location of the I/O point of the warehouse is denoted as location 0. For an item i stored next to another item k, there is an association strength A_{ik} that measures the relationship between item i and item k based on whether their relationship is classified as a strong relation extracted by the association rule mining algorithm. In the determination of the fitness of a storage location for a particular item, we define F_{ij} as the fitness measure if item i is allocated to the location j where $j \in \mathbf{L}$. The value of F_{ij} is calculated based on the following factors: First, the distance between location j and the I/O entry point (location 0), defined as d_j. Second, the strength of the association between the item and its neighbors, which suggests allocating the item close to other items with strong association. In order to define the neighbors of location j, we find all locations that are within a predetermined rectilinear distance from location j, and define the set of these locations as $\boldsymbol{n}_j$. We further determine the strength of the association of item i, as a summation of the association strength A_{ik} for all items k that are stored at locations in $\boldsymbol{n}_j$ (note: $A_{ik} \neq A_{ki}$ as association rule $i \Rightarrow k$ may not be equal to the association rule $k \Rightarrow i$, which means the association of orders that contain i always contains k may not be equal to the association of orders that contains k which also contains i. In such case, $i \Rightarrow k$ can be a strong rule while $k \Rightarrow i$ is not). Finally, we define the importance of product i based on ABC Classification scheme as w_i which the value of w_i can be determined by the past sales record according to the Pareto's Law.

3. Formulation and methodology

To formulate the fitness measure in assigning item i to location j, we define two values α and β as the importance weightings assigned to the association strength and the weighted travel distance respectively. The fitness value can be determined by the following formula:

$$F_{ij} = \alpha \sum_{k \in \boldsymbol{n}_j} A_{ik} + \beta \frac{w_i}{d_j}, \qquad (1)$$

where

k is the location defined as neighbor of location j

α and β are the weightings assigned to the strength of association factor and the weighted distance factor respectively.

w_i/d_j is the reciprocal of the weighted travel distance if product i with importance weighting w_i is assigned to be stored at location j.

The value of the weighted travel distance component w_i/d_j in (1) is proposed to allocate the popular products, defined as Class A items with larger value of w_i, to a location closer to the I/O entry point than Class C items. Even for a location with the same travel distance from the I/O entry point, the degree of fitness for a Class A item is higher than that of a Class C item. A location j with larger value of F_{ij} is considered more suitable to store item i. The values for α and β are used to normalize the effect of the two fitness components. When α is set equal to 0, the products will be allocated to a location that is nearest to the I/O entry point. This setting simulates the closest open location approach. On the contrary, if β is set equal to 0, we consider only the association of the item with the neighboring items in order to store the correlated items together, without considering the distance from the I/O entry point. One possible way to initialize the values of α and β is to set these two parameters to values such that the fitness value for an item i, with important weighting w_i, allocates to the closest location near to the depot, i.e., location j_1 at distance d_{min}, where item i has no association with any neighbor, be the same value as the same item i allocates to the farthest location j_2 at distance d_{max}, which has maximum possible associations with the neighbors, i.e., $max \sum_{k \in \boldsymbol{n}_j} A_{ik}$. With such a condition, we can determine the values of α and β by solving the following equations:

$$\beta \frac{w_i}{d_{min}} = \alpha \times max \sum_{k \in \boldsymbol{n}_j} A_{ik} + \beta \frac{w_i}{d_{max}} \qquad (2)$$

$$\alpha + \beta = 1 \qquad (3)$$

so that α and β are determined by the following formula:

$$\beta = \frac{max \sum_{k \in \boldsymbol{n}_j} A_{ik}}{w_i \left(\frac{d_{max} - d_{min}}{d_{min} \times d_{max}} \right) + max \sum_{k \in \boldsymbol{n}_j} A_{ik}} \qquad (4)$$

$$\alpha = 1 - \beta \qquad (5)$$

Based on the fitness measure, we can determine the best storage locations of all items to be put-away during replenishment.

We further define the binary decision variable x_{ij}, where

$$x_{ij} \begin{cases} 1, & \text{if item } i \text{ is allocated to store at location } j \\ 0, & \text{otherwise.} \end{cases}$$

The integer programme of the storage location assignment problem is formulated as follows:

P: Maximize $\quad \sum_{i \in P} \sum_{j \in L} F_{ij} x_{ij} \quad$ (6)

Subject to $\quad \sum_{j \in L} x_{ij} = 1 \quad$ (for all $i \in P$) $\quad$ (7)

$\quad \sum_{i \in P} x_{ij} \leq 1 \quad$ (for all $j \in L$) $\quad$ (8)

In this formulation, the objective function (6) is to maximize the total fitness of all the items to be stored in the warehouse for put-away operation. Constraint (7) indicates each item must be allocated to exactly one location in the warehouse. Constraint (8) guarantees that at most one item is allocated to each available location. In fact, if all items to be put-away are identical, this optimization problem can be solved by simply sorting the fitness measure of the locations in descending order and choosing the best p locations to be the storage locations for these p units of identical item. However, it is very common that the company replenishes several items from the supplier at the same time, and thus several items arrive at the warehouse together and needed to be put-away. In such a case, the fitness measure of an item A may not be the same as another item B to be allocated to the same location j. This problem becomes an assignment problem that is no longer trivial to solve optimally. In this paper, for simplicity, we assume the put-away operations of replenish items are performed in sequence, even though the replenishment order of several items may arrive at the same time, there is only one item to be considered on the storage location assignment for the put-away operation each time. We consider the storage location of these items by descending product importance because we believe it is more beneficial to first determine the best location for the most important items. It is also a common practice to put-away all units of the same item before putting away another item.

Routing for put-away operations

Once the storage locations of the replenish items are decided, the routing of the items can be formulated as a standard TSP problem. If the amount of items to be put-away is larger than the capacity of the routing truck, the routing problem becomes an m-TSP problem. We propose to adopt a minimum cost insertion heuristic to determine the routing of the put-away operation, because of its simple structure and easy implementation. The minimum cost insertion heuristic begins with an initial tour routed from the depot to the farthest location that stores an item decided by the storage location assignment algorithm discussed earlier, and back to the depot. Subsequent items are inserted to the route with minimum additional cost, until all items are inserted to the route. For the cases when number of items to be put-away is larger than the truck capacity, we incorporate a bidding mechanism to the insertion heuristic to determine the assignment of the items to the vehicle routes. The concept of this algorithm is based on the bidding by the available trucks which requires lowest additional cost to handle this item. This algorithm is modified based on the route construction heuristics developed by Pang (2011). The detail of the bidding algorithm is described as follows:

1. Determine the number of trucks required to perform the put-away operation for all items, by this simple equation $m = \lceil p/c \rceil$, where p is the number of items to be put-away and c is the capacity of the truck in units, assuming the items are unique in size. Now we have m trucks available to perform the put-away operations.
2. Identify the item location that is farthest away from the I/O point and assign it to the 1st truck route. The remaining $\text{m} - 1$ truck routes are initialised with zero travel distance.
3. For each of the m truck routes, determine the best location to insert an item with minimum additional cost to the current truck routes. If a truck is loaded with full capacity, stop considering that truck for further item insertion.
4. The truck route which requires minimum additional cost for inserting an item wins the bid; the item is assigned to that truck.
5. Repeat Step 3 and 4 until all items are assigned to the trucks.

The implementation of the above insertion bidding algorithm is presented through the pseudo-code shown below.

Inputs:

p = number of items to be put-away;
c = capacity of a truck (in unit);
i_farthest = an item to be put-away located farthest from the I/O entry point.

Procedure min_bid_insert

$m = \lceil p/c \rceil$;
$current_{item} = 1$;
assign $i\_farthest$ to the *1[st]* truck route;
while ($current_{item} \leq p$) {
for $v = 1 \; to \; m$ {
find the best position to insert $current_{item}$ to the route of truck v which has available capacity;
} // end for
assign the $current_{item}$ to the truck route with minimum insertion cost;
update the available capacity of the truck;
$current_{item} = current_{item} + 1$;
} // end while.

Decisions on order-picking operations

For the order-picking operation, there are two decisions that need to be made. Firstly, which item in the warehouse should be selected for fulfilling the customer order? In some situations, this decision can be affected by the first-in-first-out (FIFO) policy, especially for those items with specific product life, i.e., perishable goods. In this paper, we assume the adoption of the FIFO policy because we believe that FIFO policy is a common practice adopted for managing the products within the warehouse in real situations. However, we assume the items that are replenished together at one time have same priority for picking, thus only the same items from different replenishment orders are considered in the FIFO policy. Secondly, how to route the order picker truck to visit the storage locations and collect all selected items stored in the warehouse? The main objective of order-picking operation is to minimize the travel distance and/or lead time of picking all the items, so as to respond to the customer orders faster. In this paper, we propose to make these two decisions simultaneously, by determining the product location to pick the items and constructing the route iteratively. Since the main focus of this paper is on the determination of the storage location of the correlated products, but not the routing of the order-picking operations, we propose to adopt a revised heuristic based on the minimum cost insertion algorithm used in the routing of put-away operation described earlier in this paper. In fact, it is already a difficult problem to determine the optimal solution of selecting the best pick locations to satisfy a customer order.

When constructing the order picking route, we start with an item of an order that has the least quantity (say, item Z) stored in the warehouse, and construct the route that visits the locations of all remaining items of the order iteratively. This is because least quantity of item Z are stored in the warehouse, so that the number of routes to be generated for final selection is minimized when compared with the routes constructed with the most popular product. Once all possible routes of picking the order are generated, we select the route which requires the shortest total travel distance as the final order-picking route for this customer order. The detail of the routing algorithm of the order-picking operation is described as follows:

1. Identify the item on a customer order that has the least quantity stored in the warehouse. Define this item as the Least Available Product (LAP).
2. Select the LAP located closest to the I/O point, and construct the order-picking route.
3. Build an initial route that picks up the LAP and return to the I/O point.
4. Insert the remaining items on this customer order to the route by minimum cost insertion heuristics.
    - 4.1 Identify an item on the pick list that is stored at a location in the warehouse which requires minimum additional travel distance on the current route.
    - 4.2 If there is more than one item with the same minimum insertion cost identified, pick the item with smaller quantity available and insert it to the route.
    - 4.3 If the same insertion cost is found for multiple items, choose the item with larger value of association with its neighbors and insert it to the route.
    - 4.4 Update the travel distance of the route.
    - 4.5 Compare the travel distance of this route with the total travel distance of the best route. If the route length is already longer than that of the best route, stop constructing the route as this route is not better than the existing best solution. Otherwise, repeat Step 4 until all items on the customer order are inserted to the route.
    - 4.6 If the total travel distance is shorter than the best solution, update the new best solution.
5. Repeat Step 3 and Step 4 with a new LAP until all LAPs' routes are constructed and compared.
6. Select the best route which has the shortest travel distance.

The pseudo-code of the order-picking heuristic implementation is presented as shown below.

Inputs:

$$\mathbf{P} = \ set\ of\ items\ on\ a\ customer\ order;$$
$$\mathbf{V} = \emptyset; set\ of\ items\ assigned\ to\ a\ truck;$$

Procedure order-picking_route

$current\_distance = \infty$;

$min\_distance = \infty$;

locate the *Least Available Product* (LAP) in **P**, determine the quantity available in the warehouse, say there are k units identified;

sort these k units of the LAP by their distances to the I/O entry point in ascending order;

for $r = 1$ to sorted(k) {

construct an initial route to pick the r^{th} LAP;

update $current\_distance$;

add the r^{th} LAP to **V** and remove it from **P;**

while ($\mathbf{P} \neq \emptyset$){

identify an item in $\mathbf{P}$ that requires minimum cost to insert to the current route;

add the item to **V** and remove it from **P;**

update $current\_distance$;

if ($current\_distance > min\_distance$) {

Stop! This route is not better than the best route;

}; // end if

}; // end while

if ($r = 1$) { // This is the first route.

$min\_distance = current\_distance$;

};

Else if ($current\_distance < min\_distance$) { // If the new route length is shorter.

$min\_distance = current\_distance$;

$best\_route = r$;

}; // end if

}; // end for

4. Computational experiments

In order to evaluate the performance of our proposed association rule based storage location assignment algorithm, we conduct extensive computation experiment that simulates the warehouse operations of real life applications in different scales to analyse the total travel distance for both put-away and order-picking operations in a randomized warehouse. We compare the performance of our proposed algorithm with the closest open location policy and purely dedicated policy which these policies are commonly adopted in real practices. For the design of the computational experiment, we define the parameters on the warehouse layout as presented in Section 2.

We further introduce the following assumptions for the experimental study:

- Each put-away or picking tour begins and ends at the I/O entry point of the warehouse.
- The shelves in the warehouse are two-sided so that products can be stored and picked on both sides with single-block storage layout as shown in Figure 1.

- Each item occupies exactly one storage space and all storage spaces are of the same size, i.e., 1 metre wide.
- The warehousing operation is based on discrete order-picking with multi-command cycle policy so that order-picking or put-away operation of multiple items is carried out in one trip.
- Only horizontal travel distance is considered for the performance measure. For the vertical movement up or down the shelves to different levels for picking or put-away operation, we can easily extend the model to cover this scenario by modifying the distance calculation between the storage locations in a 3-dimensional space.
- The distance defined as neighbor of a location is set equal to two metres in rectilinear distance away from that location.
- In case the available amount of stock in the warehouse is not enough to fulfil the incoming customer order, a replenishment order is placed. The reorder quantity for replenishment is determined by the popularity of the item to be replenished. We also assume there is no lead time for receiving the replenish items, i.e., the required products are replenished spontaneously as this is not the focus of our study.
- When more than one product are reordered, products with higher popularity will first be put-away to the warehouse so that they are stored nearer to the I/O entry point.

In addition to the above assumptions, the characteristics of the order history and the mechanism for correlation generation among products are as described below:

Synthetic transaction data is generated using the ARtool platform developed by Cristofor (2002) and it is also used as the customer order database generator. The information of the correlation among products is useful for assigning these products to locations nearby in order to reduce the order-picking distance. Since the focus of this research is on the storage allocation policy that utilizes the correlation relationship information among the products, but not the efficiency of the association rules extraction algorithm. Thus, we do not consider the efficiency of the data mining algorithm used. Thus, in this research study, we employ the Apriori algorithm proposed by Agrawal *et al*. (1993) for the association rules mining in the software package because it is more commonly used and efficient in generating all strong associations. In fact, the ARtool has the option to choose other algorithms for association rules extraction, such as the FPgrowth algorithm developed by Han *et al.* (2000) and the CoverRules algorithm developed by Cristofor and Simovici (2002). The details of mining association rules by the Apriori algorithm can be found in Agrawal *et al*. (1993), Srikant and Agrawal (1997), and Han and Kamber (2001).

The order history generated is also used to determine the popularity of the products based on the frequency of occurrence of an item in the orders. The objective of knowing the popularity of the products is to identify the most frequently ordered items and allocating them in strategic locations, i.e., location closer to the I/O entry point, so that the distance for both put-away and order-picking operations is reduced.

Meanwhile, the setting on number of stock-keeping units (SKUs), number of customer orders and number of items requested in each order for the synthetic data generation will be analyzed as parameters in the computational experiments. The experiments were conducted on a PC with a 2.93-GHz processor with 16 GB memory.

4.1 Data generation

The experiment data generation is based on the operations of a computer and networking device spare-part warehouse. In order to simulate difference scales of the operations, three warehouse sizes with M = 30, 40 & 50 are tested, i.e., 30x30, 40x40 and 50x50 as the number of shelves and number of storage spaces on each side of the shelves. In order words, there are $30\text{x}30\text{x}2 = 1{,}800$, $40\text{x}40\text{x}2 = 3{,}200$ and $50\text{x}50\text{x}2 = 5{,}000$ storage spaces available in each of the defined warehouse sizes respectively. In the initialization stage, we use the ARtool software to generate 2000 synthetic customer orders by specifying the number of SKUs, number of items on an order, number of patterns for correlated products, etc. With this 2000 synthetic orders, we randomly pick 1000 orders as the order history for association rule extraction and popularity calculation, the other 1000 orders are used as the order-picking and replenishment process simulation. Products are initially allocated to the storage location according to their popularity to fill up the warehouse based on the practice of the respective storage allocation approaches, i.e., closest open location, dedicated and correlated storage. After that, 1000 customer orders are handled and picked for the performance comparison of different approaches, replenishment orders as described above are issued when necessary.

Besides the parameters on the warehouse size, we consider three scenarios on different combinations of number of Stock-Keeping Unit (SKU) and number of items in each customer order. For the small and medium size warehouses (i.e., 30×30 & 40×40), we run the experiment with 20, 30, 40 SKUs and an average of 15 items on each customer order. Note that several units of the same SKU may be requested on one order, such that each order has $\mathrm{N} \leq 15$ unique items. Similarly, with the same number of SKUs, an average of 20 items in an order are generated for the medium and large size warehouses (i.e., 40×40 & 50×50). Furthermore, in order to test the reliability of our proposed approach, ten random samples with different combinations of α and β values from $0 - 1$ with 0.1 increment are randomly generated for performance comparison with the other two classical storage assignment policies, namely closest open location and purely dedicated storage policies. In total, $2 \times 2 \times 3 \times 10 = 120$ sets of data are generated and tested, which simulate different scenarios for the analysis of the potential cost saving attainable by using the proposed storage location assignment algorithm. A flowchart that illustrates the overall operations flow of the simulation model for the experimental study is presented in Figure 2.

4.2 Computational results

The experimental results of each storage allocation approach are presented in Table 1. Our proposed Association Rule Based storage location assignment algorithm, which is called ARB algorithm, was compared with the closest open location (COL) policy and purely dedicated (Dedicated) policy, which the closest open location policy assigns the replenish products to the empty locations nearest to the I/O entry point and the purely dedicated approach stores the replenish products to locations that were pre-assigned specifically to the class of the replenish products based on its popularity. In Table 1, the result shows that the distance traveled for the order-picking operation (Column C5) contributes significantly to the overall cost of warehousing operation, it accounts for about 45~77% of the total travel distance which varies with the storage allocation approaches adopted. In terms of total distance traveled (C4), our proposed ARB approach requires shorter distance when compared with the closest open location policy, ranging from 11% to 37% saving. The saving on total traveled distance is much significant when compared with the dedicated storage approach, ranging from 35% to 61% saving. If considering only the order-picking distance (C5), the ARB algorithm can save even more on average at 56.3% and 76.8% when compared with the closest open location policy and dedicated approach respectively. This saving is tremendous if we agree that order-picking operation contributes significantly to the warehousing expenses. The saving for the cases in larger warehouse sizes is much obvious, while number of SKUs does not significantly affect the performance of our proposed storage allocation algorithm.

For the travel distance of the put-away operations (C6), it is expected that ARB approach loses its advantages when compared with the closest open location policy. This is because the closest open location policy aims at storing the replenish products at the empty locations that are nearest to the I/O entry point. Thus, such policy results in shorter travel distances is not surprising. In contrast, the ARB algorithm intentionally stores the products at locations deeper inside the warehouse, but nearer to its correlated products which facilitates faster order-picking operations for satisfying customer orders, so that the put-away travel distance is longer. For the dedicated approach, it is slightly advantageous in shorter replenishment travel distance over the ARB algorithm for the small size warehouse. However, there are significant inefficiencies on the dedicated system for the cases in large warehouse size and large number of SKUs, which accounts for 14.1% for the cases with 50 x 50 warehouse size and 28.1% for the cases with 40 x 40 warehouse size both with 40 SKUs and 20 items in an order. The possible reason for this observation on the dedicated approach may due to the fact that some items are classified as non-frequently purchased items, which are assigned to be stored deep inside the warehouse according to the ABC classification scheme. When these items are replenished, the worker needs to travel to the far deep corner of the warehouse in order to put-away the items to the storage locations.

In terms of computation time (C7), our proposed ARB algorithm obtains the result in shortest time. Since there are more order-picking decisions to be made than the put-away operations, it is an advantage of using the ARB algorithm to group correlated products together when replenishing the products, such that it is faster for the order-picking system to determine which products should be picked to satisfy a customer order. As a result, the computing time for ARB approach is shorter than the other two approaches by 12.3% and 38.1% on average. Overall, all three approaches can obtain the storage location assignment decision within reasonable time so that they are suitable to be implemented as real time decision making algorithms in warehousing management. The tabulated results shown in Table 1 are presented graphically in Figure 3 for better visualization.

For the reliability analysis, the result shown in Figure 4 reveals that ARB algorithm outperforms the other two storage approaches, regardless of the setting on the α and β values. It jumps to much smaller total travel distance once the associations among products are considered. The experiment results show that the order-picking distance is reduced if correlated products are stored nearby when replenishing these products. This observation aligns with our initial objective to develop the storage allocation decision algorithm which considers the correlation among products.

4.3 Extended computational experiment

In order to further evaluate the proposed storage allocation algorithm on its application to warehouses with larger number of SKUs, we perform another set of experiments with parameters that customer orders request more items and the warehouse size is bigger. We consider the warehouse sizes with M=50 and 60, i.e., 50 x 50 & 60 x 60. There are 80 SKUs and 150 SKUs for the cases with warehouse sizes of 50 x 50 and 60 x 60 respectively. Other parameters are the same as the setting of the first experiment described, except with $\alpha = 0.2$ and $\beta = 0.8$.

The result of the extended experiment is presented in Table 2. The 4^{th} to 6^{th} columns show the total travel distance including both put-away and order picking operations. It reveals similar observations as the first experiment that number of SKUs does not affect the performance of our proposed allocation algorithm in terms of saving on total travel distance when compared with the closest open location policy and purely dedicated approach. Instead, the size of the customer orders affects significantly on the saving that more saving is obtainable when customers order less number of items. The saving on total travel distance drops from around 15% to only 2% when customers order 20 items are increased to 40 items. If only the order picking distance is considered, the saving drops from around 30% to only 6% correspondingly. This observation is explainable that when a customer order has more items to be picked, the chance that it contains some un-correlated items will be high so that the order picker has to pick up those items at farther locations which are not near to those correlated items stored together. This situation significantly increases the order picking distance for our proposed storage allocation policy.

Similar to the result of the first experiment, the put-away distance for the ARB approach is longer than that of the closest open location policy, which is also well expected and explained in the previous section. In fact, it is interesting to note that the size of the warehouse does not have any obvious impact on the performance of our proposed algorithm, in terms of the percentage saving of the travel distance. It only affects the total distance travelled by the order picker for both put-away and order picking operations, longer travel distance is required for handling the put-away and order picking orders for a bigger warehouse. This result is also aligned with the observations obtained in the first experiment. We believe this research topic is worthy of further investigation as customer order data are more readily available as the information technology advances. The management of the warehouse should make best use of this information to enhance productivity and responsiveness in reacting to customers timely needs.

5. Conclusion

In this paper, we present a storage allocation algorithm that utilizes data mining technique to extract the relationship among products on customer orders that determines the storage location of these items in a randomized warehouse. The logic of the proposed storage approach is to optimize the order-picking activity, without sacrificing too much on the put-away operations as correlated products are stored nearby that the distance of the storage location from the I/O entry point of the warehouse is also considered, aiming at minimizing the manual efforts in the warehousing operations by reducing the total distance traveled for both put-away and order-picking operations.

The performance and effectiveness of the proposed methodology are examined by measuring the total travel distance for both order-picking and put-away operations and comparing the result with closest open location policy and dedicated storage approach. The results show that our proposed storage approach is effective in improving the overall warehousing operating efficiency when compared with the other two approaches. This storage approach is especially useful when customers order associated items together with small order size such that the order picker only require to travel through a small region of the warehouse in order to pick the items of that order. The findings of this study are particularly beneficial to the warehouse operators in Hong Kong as the warehouse rent is high and the space availability is limited, better storage assignment policy in a randomized warehouse can both optimizes the space utilization and the cost of warehousing operations.

However, this study has several limitations which worth our attention to further develop a better strategy for managing the warehouse. Firstly, the data for the experiment are generated synthetically by using the online software tool, i.e., ARTool, instead of using the real application data. It is because the problem scale of the real-life application in the computer spare parts warehouse is too small for any meaningful analysis on the performance comparison. However, it triggered our attention to study such a storage allocation problem which considers the correlation among items stored in a warehouse when customers always order similar items together. The experiments that are performed in this paper simulate different scales of warehousing operations with sizes ranging from 1800 to 7200 unit spaces. Besides, our proposed algorithm has some limitations that can be investigated for possible future extension. For instance, the routing decision for both put-away and order-picking operations can be further examined. One possible alternative for improving the routing decision is to convert the problem to a Traveling Salesmen Problem (TSP) or Vehicle Routing Problem (VRP) that minimizes the total travel distance. With the TSP/VRP formulation, algorithms with reported efficiency can be adopted in order to obtain better result with shorter total travel distance, which further enhance the operation efficiency of the warehouse. However, computing time is an issue that should be considered when adopting the algorithm to solve these routing related problems. Finally, an effective order history updating mechanism should also be incorporated to the algorithm in order to capture the changing trend on customer preferences. This extension can help enhance the system to dynamically determine the storage location of the replenish items, as some items are becoming more popular while the sales volume of some other items are dropping towards the end of their product life. All in all, employing the concept of data mining in different applications is a promising area for research as data is more readily available and information technology has becoming more powerful and affordable. It is expected that big data will continue to be a popular topic in different contexts for the research community.

ACKNOWLEDGMENTS

This research work was supported in part by the Research fund PolyU 5244/11E of The Hong Kong Polytechnic University. The authors would also like to thank the anonymous reviewers for the constructive comments which make this research paper better in quality and presentation.

References

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *ACM SIGMOD Conference*, Washington DC, USA, 254-259.

Agrawal, R. & Srikant, R. (1994). Fast Algorithms for Mining Association Rules. *In Proceedings of the 20th International Conference on Very Large Data Bases (VLDB' 94)*, Santiago, Chile, 487-499.

Battini, D., Calzavara, M., Persona, A., & Sgarbossa, F.(2015) Order picking system design: the storage assignment and travel distance estimation (SA&TDE) joint method, *International Journal of Production Research*, 53:4, 1077-1093.

Bindi, F., Manzini, R., Pareschi A., & Regattieri, A. (2008). Similarity-based storage allocation rules in an order picking system: an application to the food service industry. *International Journal of Logistics Research and Applications,* 12(4): 233-247.

Chan, H.L., Pang, K.W., & Li, K.W. (2010). Association rule based approach for improving operation efficiency in a randomized warehouse. *In Proceedings of the 2011 International Conference on Industrial Engineering and Operations Management,* Kuala Lumpur, Malaysia.

Chen, M.C., Huang, C.L., Chen, K.Y., & Wu, H.P. (2005). Aggregation of orders in distribution centres using data mining. *Expert Systems with Applications*, 28(3):453-460.

Chiang, D.M.H., Lin, C.P., & Chen, M.C. (2008). A product allocation approach based on association rule mining. *In Proceedings of the 9th Asia Pacific Industrial Engineering and Management Systems Conference*, 2418-2423.

Chiang, D.M.H., Lin, C.P., & Chen, M.C. (2011) The adaptive approach for storage assignment by mining data of warehouse management system for distribution centers, *Enterprise Information Systems*, 5:2, 219-234

Chuang Y.F., Lee, H.T., & Lai, Y.C. (2012) Item-associated cluster assignment model on storage allocation problems. *Computers & Industrial Engineering*, 63(4), 1171–1177.

Cristofor, L., & Simovici, D. (2002) Generating an informative cover for association rules. *Computer Science Department Technical Report TR-02-01*, UMass/Boston.

Elsayed, E.A. (1981). Algorithms for optimal material handling in automatic warehousing systems. *International Journal of Production Research*, 19(5):525-535.

Elsayed, E.A. & Stern, R.G. (1983). Computerised algorithm for ordering in automated warehousing systems. *International Journal of Production Research*, 21(4):579-586.

Elsayed, E.A. & Unal, I.O. (1989). Order batching algorithms and travel time estimation for automated storage/retrieval systems. *International Journal of Production Research*, 27(7):1097-1114.

Frazelle, E.A. & Sharp, G.P. (1989). Correlated assignment strategy can improve any order-picking operation. *Industrial Engineering*, 21(4):33-37.

Frazelle, E.H. (1990). Stock Location Assignment and Order Batching Productivity. *Ph.D. Thesis*. Georgia Institute of Technology, Atlanta, Georgia.

Gagliardi, J-P., Renaud, J. & Ruiz, A. (2012) On storage assignment policies for unit-load automated storage and retrieval systems, *International Journal of Production Research*, 50(3): 879-892.

Gibson, D.R. & Sharp, G.P. (1992). Order batching procedures. *European Journal of Operational Research*, 58(1): 57-67.

Glock, C.H. & Grosse, E.H. (2012) Storage policies and order picking strategies in U-shaped order-picking systems with a movable base, *International Journal of Production Research*, 50(16): 4344-4357.

Gu, J. (2005). The forward reserve warehouse sizing and dimensioning problem. *Ph.D. Thesis*. Georgia Institute of Technology, Atlanta, GA.

Guo, X., Yu, Y., & De Koster, R. (2015): Impact of required storage space on storage policy performance in a unit-load warehouse, *International Journal of Production Research*, forthcoming.

Han, J. & Kamber, M. (2001). *Data mining: Concepts and techniques*. San Francisco: Morgan Kaufmann.

Han, J., Pei, J., & Yin, Y. (2000). Mining Frequent Patterns without Candidate Generation. *Proceedings of ACM-SIGMOD International Conference on Management of Data*, 1-12.

Helsgaun, K., (1998) An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic. *DATALOGISKE SKRIFTER (Writings on Computer Science)*, No. 81, Roskilde University.

Helsgaun, K., (2015) Solving the Equality Generalized Traveling Salesman Problem Using the Lin-Kernighan-Helsgaun Algorithm. *Mathematical Programming Computation*, 7(3):269-287.

Hwang, H. & Lee, M.K. (1988). Order batching algorithms for a man-on-board automated storage and retrieval system. *Engineering Cost and Production Economics*, 13:285-294.

Hwang, H., Baek, W., & Lee, M.K. (1988). Clustering algorithms for order picking in an automated storage and retrieval system. *International Journal of Production Research*, 26(2):189-201.

Kim, K.H. (1993). A joint determination of storage locations and space requirements for correlated items in a miniload automated storage-retrieval system. *International Journal of Production Research*, 31:2649-2659.

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.

Li, J., Moghaddam, M., & Nof, S.Y. (2015) Dynamic storage assignment with product affinity and ABC classification — a case study. *International Journal of Advanced Manufacturing Technology*, forthcoming.

Linn, R.J. & Wysk, R.A. (1987). An analysis of control strategies for automated storage and retrieval systems. *INFOR*, 25:66-83.

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

Malmborg, C.J. (1996). Storage assignment policy tradeoffs. *International Journal of Production Research*, 34:363–378.

Meneghetti, A. & Monti, L. (2014) Multiple-weight unit load storage assignment strategies for energy-efficient automated warehouses, *International Journal of Logistics Research and Applications*, 17(4): 304-322

Pang, K.W. (2011). An adaptive parallel route construction heuristic for the vehicle routing problem with time windows constraints. *Expert Systems with Applications*, 38:11939-11946.

Quintanilla, S., Pérez, Á., Ballestín, F., & Lino, P. (2015) Heuristic algorithms for a storage location assignment problem in a chaotic warehouse, *Engineering Optimization*, 47(10), 1405-1422.

Rao, S.S. & Adil, G.K. (2013) Class-based storage with exact S-shaped traversal routeing in low-level picker-to-part systems, *International Journal of Production Research*, 51(16), 4979-4996.

Roodbergen, K.J. & De Koster, R. (2001). Routing order-pickers in a warehouse with a middle aisle. *European Journal of Operational Research,* 133(1), 32–43.

Rosenwein, M.B. (1996). A comparison of heuristics for the problem of batching orders for warehouse selection. *International Journal of Production Research*, 34(3), 657-664.

Srikant, R. & Agrawal, R. (1997). Mining generalised association rules. *Future Generation Computer Systems*, 13(2-3):161-180.

Tompkins, J.A., White, J.A., Bozer, Y.A., & Tanchoco, J.M.A. (2003). *Facilities Planning*. John Wiley & Sons, New Jersey.

Van den Berg, J.P. & Zijm, W.H.M. (1999). Models for warehouse management: Classification and examples. *International Journal of Production Research,* 59, 519–528.

Xiao, J. & Zheng, L. (2009). A correlated storage location assignment problem in a singleblock- multi-aisles warehouse considering BOM information. *International Journal of Production Research*, 48(5), 1321-1338.

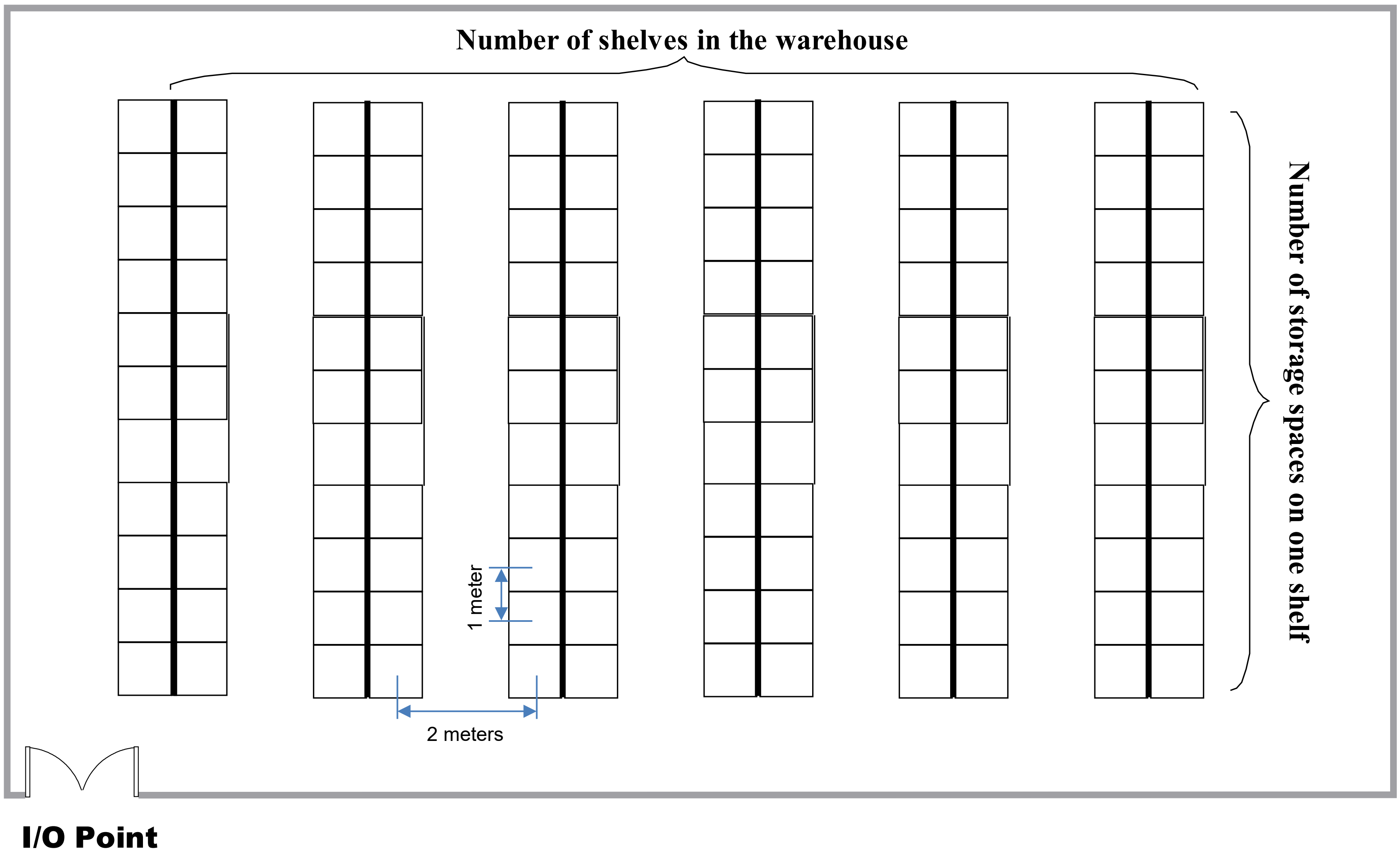

Figure 1. Layout setting of the warehouse

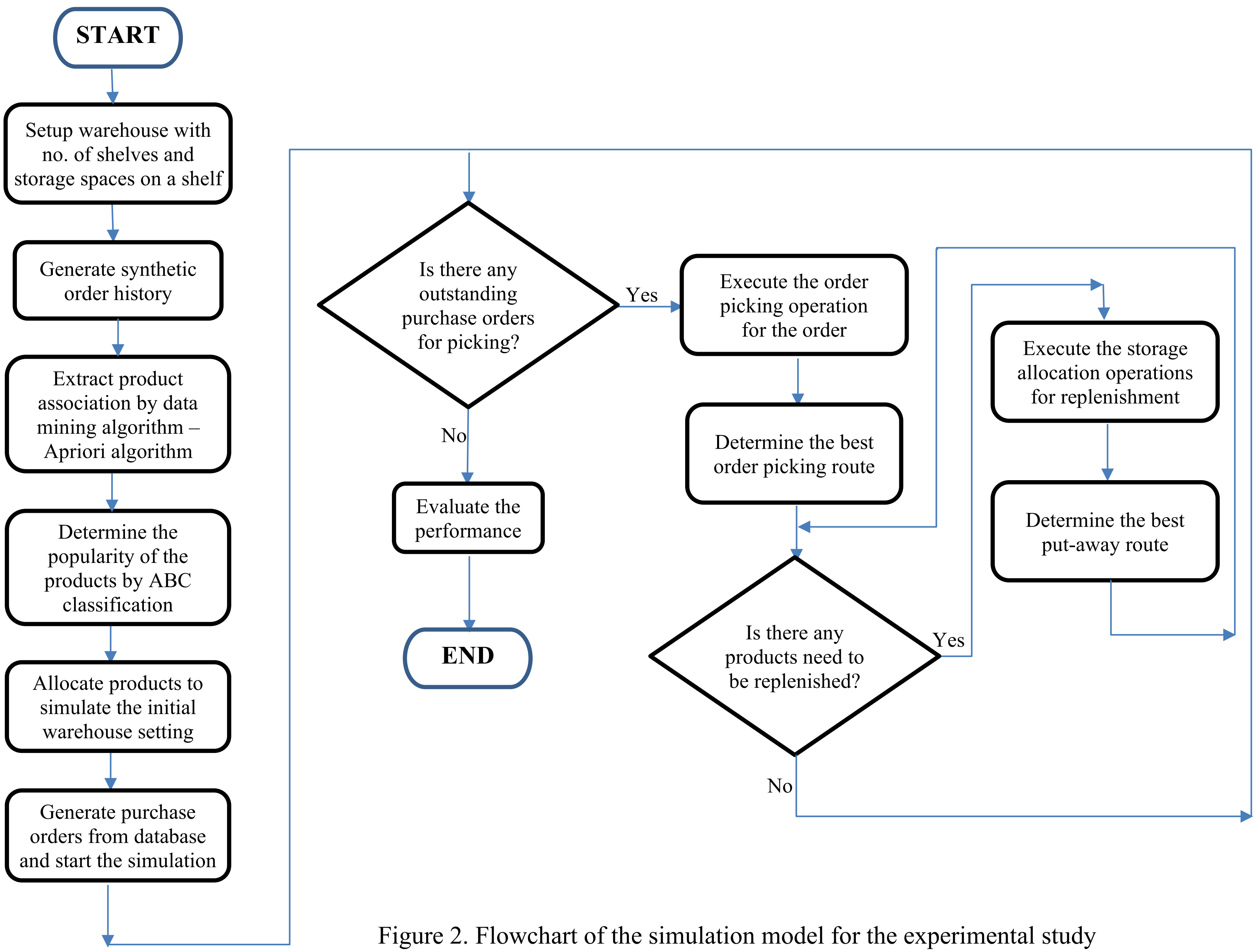

Figure 2. Flowchart of the simulation model for the experimental study

Table 1: Result comparison of the proposed storage allocation approach

C1	C2	C3	C4			C5			C6			C7		
No. of Shelves X Storage spaces	No. of SKUs	No. of items in an order	Total Distance (meter)			Order-Picking Distance (meter)			Put-away Distance (meter)			Computing Time (sec)		
			ARB[#]	COL*	Dedicated	ARB	COL	Dedicated	ARB	COL	Dedicated	ARB	COL	Dedicated
30 X 30	20	15	226371	275716 21.8%	311896 37.8%	123324	188484 52.8%	217408 76.3%	103047	87232 -15.3%	94488 -8.3%	1371	1513 10.4%	1937 41.3%
30 X 30	30	15	224869	273286 21.5%	313281 39.3%	123802	186142 50.4%	218649 76.6%	101067	87144 -13.8%	94632 -6.4%	1439	1491 3.6%	1871 30.0%
30 X 30	40	15	224949	269629 19.9%	312383 38.9%	121766	182602 50.0%	213264 75.1%	103183	87027 -15.7%	99119 -3.9%	1389	1491 7.3%	1694 22.0%
40 X 40	20	15	249424	330688 32.6%	400448 *60.5%*	152753	249037 63.0%	307938 101.6%	96671	81651 -15.5%	92510 -4.3%	4298	4791 11.5%	6680 55.4%
40 X 40	30	15	247663	339467 37.1%	384825 55.4%	154206	255834 65.9%	288916 87.4%	93457	83633 -10.5%	95909 2.6%	4291	4839 12.8%	5218 21.6%
40 X 40	40	15	250458	331408 32.3%	384010 53.3%	154960	248582 60.4%	288110 85.9%	95498	82826 -13.3%	95900 0.4%	4155	4756 14.5%	5176 24.6%
40 X 40	20	20	325221	366858 12.8%	441447 35.7%	160749	233997 45.6%	266537 65.8%	164472	132861 -19.2%	174910 6.3%	5332	5464 *2.5%*	6143 15.2%
40 X 40	30	20	348253	386403 *11.0%*	490363 40.8%	159533	230555 *44.5%*	263142 64.9%	188720	155848 -17.4%	227221 20.4%	5072	5794 14.2%	6623 30.6%
40 X 40	40	20	365940	413675 13.0%	551953 50.8%	161867	241674 49.3%	268181 65.7%	204073	172001 -15.7%	283772 39.1%	5274	5901 11.9%	7035 33.4%
50 X 50	20	20	364416	452383 24.1%	490673 34.6%	190270	315543 65.8%	331903 74.4%	174146	136840 -21.4%	158770 -8.8%	12324	13647 10.7%	16253 31.9%
50 X 50	30	20	399689	498919 24.8%	551468 38.0%	200667	331506 65.2%	351692 75.3%	199022	167413 -15.9%	199776 0.4%	11551	15043 30.2%	22780 97.2%
50 X 50	40	20	409911	498994 21.7%	585056 42.7%	194389	317021 63.1%	334257 72.0%	215522	181973 -15.6%	250799 16.4%	12324	14508 17.7%	18923 53.5%

[#]ARB – Our proposed Association-Rule Based (ARB) storage allocation algorithm.

*COL – Closest Open Location (COL) storage allocation policy.

Figure 3. Performance comparison of the proposed association-rule based storage allocation algorithm.

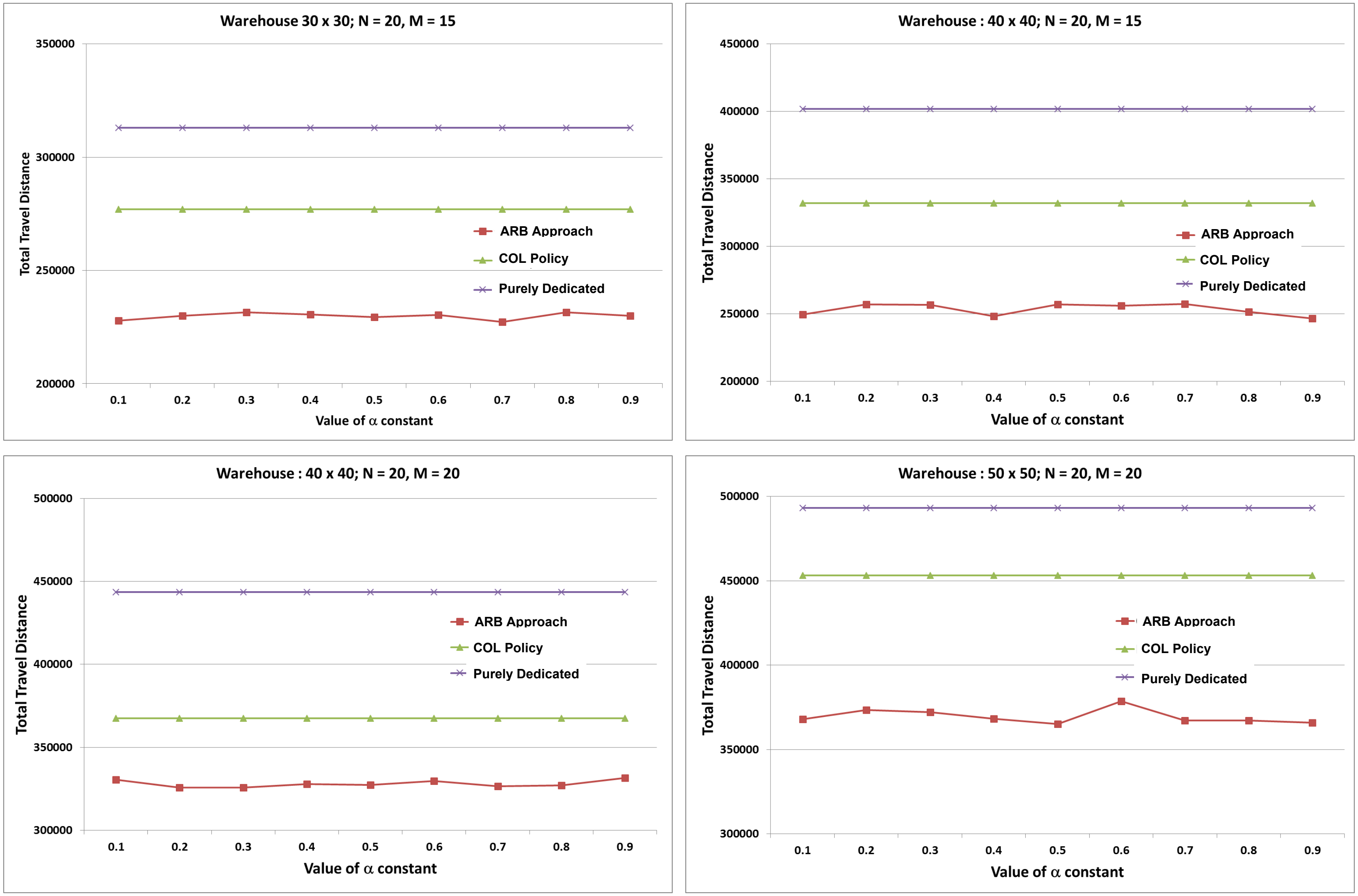

Figure 4a. Sensitivity analysis of our proposed algorithm on α value for the cases with N = 20.

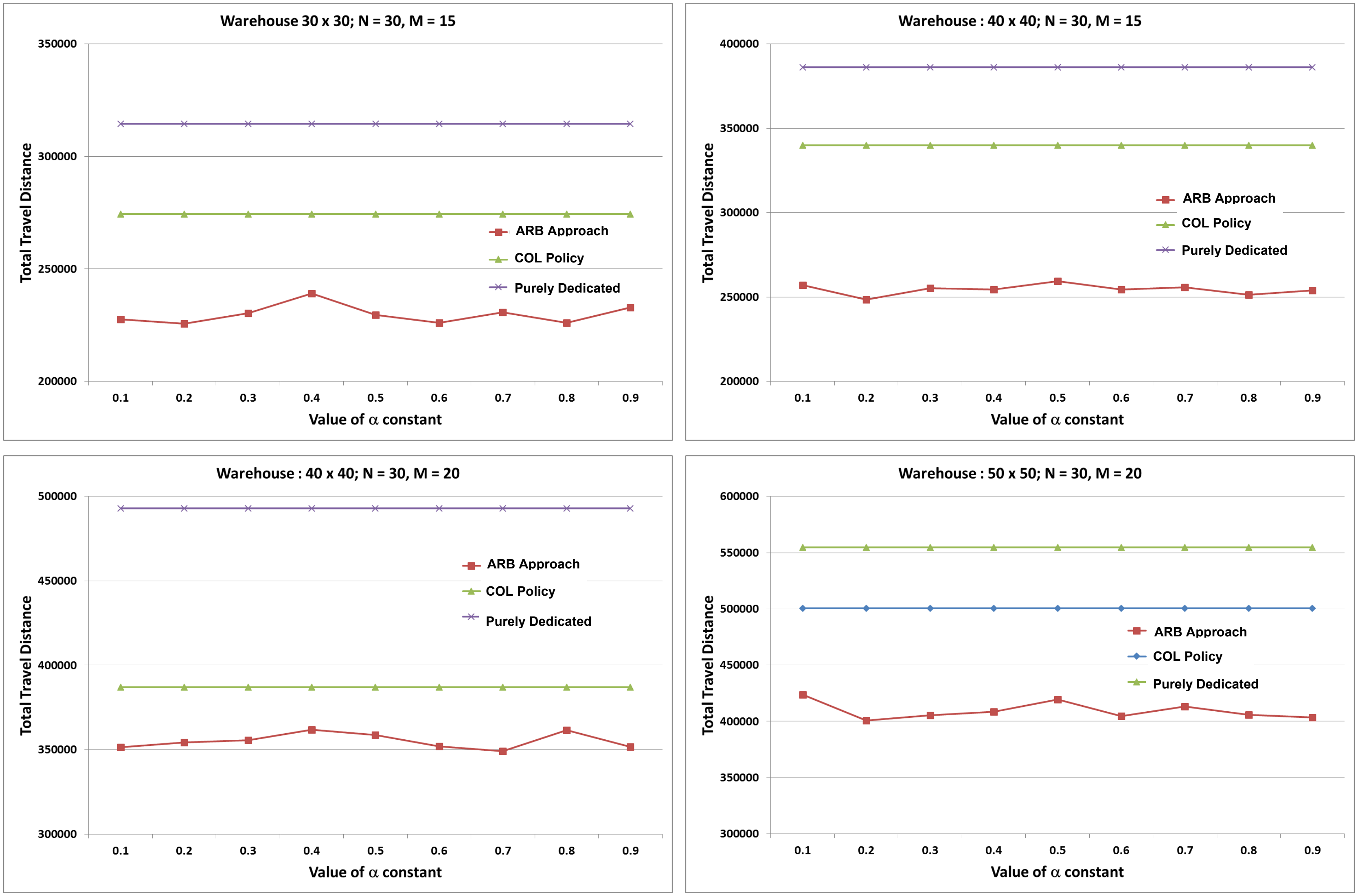

Figure 4b. Sensitivity analysis of our proposed algorithm on α value for the cases with N = 30

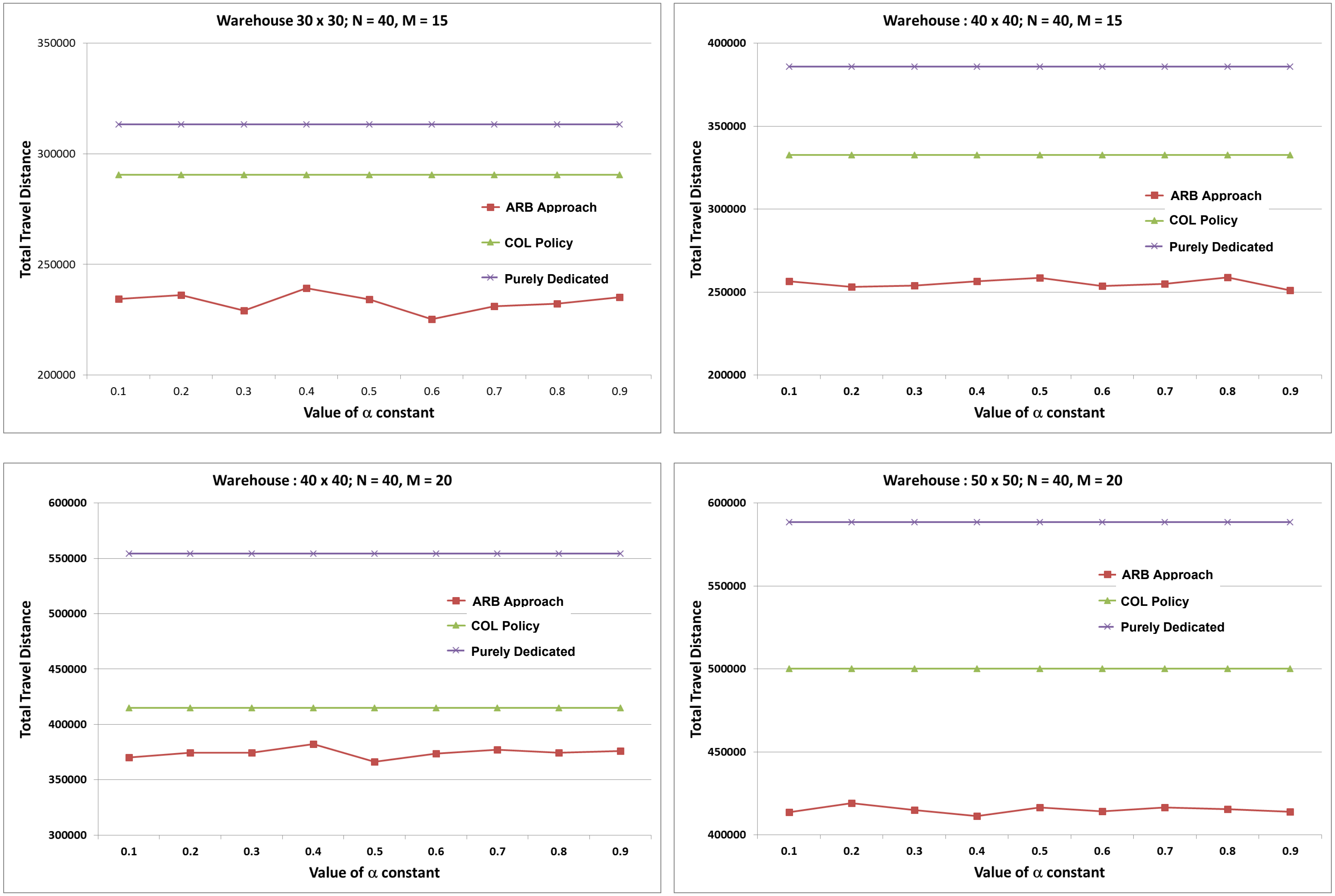

Figure 4c. Sensitivity analysis of our proposed algorithm on α value for the cases with N = 40

Table 2: Result comparison of the proposed storage allocation approach on large scale warehouses

No. of Shelves X Storage spaces	No. of SKUs	No. of items in an order	Total Distance (meter)			Order-Picking Distance (meter)			Put-away Distance (meter)			Computing Time (sec)		
			ARB	COL	Dedicated	ARB	COL	Dedicated	ARB	COL	Dedicated	ARB	COL	Dedicated
50 X 50	80	20	478845	559382 14.4%	791182 39.5%	221369	333545 33.6%	362488 38.9%	257476	225837 -14.0%	428694 39.9%	12711	13957 8.9%	18489 31.3%
50 X 50	80	30	876488	899574 2.6%	1139677 23.1%	337288	398774 15.4%	429823 21.5%	539200	500800 -7.7%	709854 24.0%	18128	18556 2.3%	24693 26.6%
50 X 50	80	40	1299617	1318978 1.5%	1421465 8.6%	730657	778018 6.1%	756020 3.4%	568960	540960 -5.2%	665445 14.5%	27035	23952 -12.9%	44260 38.9%
60 x 60	150	20	604365	708191 14.7%	1220372 50.5%	301216	425997 29.3%	477259 36.9%	303149	282194 -7.4%	743113 59.2%	18035	18517 2.6%	22754 20.7%
60 x 60	150	30	1224231	1269748 3.6%	1723235 29.0%	525264	583375 10.0%	681410 22.9%	698967	686373 -1.8%	1041825 32.9%	13868	11031 -25.7%	13496 -2.8%
60 x 60	150	40	1810355	1847524 2.0%	2252110 19.6%	713490	759206 6.0%	771308 7.5%	1096865	1088318 -0.8%	1480802 25.9%	15187	11665 -30.2%	11696 -29.8%