A Data-Intensive Analysis Augmented Simulation Model of a Distribution Center Operations

Jesus A. Jimenez [1], Francis A. Méndez Mediavilla [2], Cecilia Temponi [2]

[1] Ingram School of Engineering, Texas State University, San Marcos TX, USA
[2] McCoy College of Business, Texas State University, San Marcos TX, USA
{jj30@txstate.edu, fm16@txstate.edu, ct01@txstate.edu}

Abstract. This study presents the application of both data-intensive analysis and simulation to improve the operations of a distribution center in the food retail industry. This paper shows that creating families of stock keeping units (SKU's) frequently found in the same order, as well as positioning an item near the rest of the members of the family can improve the performance of the distribution center's (DC) order picking time – a key performance indicator of the facility's efficiency. The families of frequently ordered SKU's are the result of applying association rules mining algorithms, which extract family formation criteria from a big dataset, i.e. SKU's per order, order quantity per SKU, product weight, etc. This algorithm uses three metrics from the frequent rules mining literature: *support*, *confidence*, and *lift,* which control not only the quality of the families formations, but also the algorithm's computing time. A simulation model, built in the WITNESS® simulation language, represents the operations of a virtual distribution center, and is therefore used to test the different positioning configurations of SKU's in the distribution center. Furthermore, the simulation model accounts for other DC dynamics, such as aisle flow distribution and vehicle congestion. Results show that the layout configuration generated from the association rules mining algorithm improves the baseline layout configuration by 12 percent.

Keywords: Simulation, Data-Intensive Analysis, Supply Chain, Distribution Centers, Facilities Layout.

1 Introduction

In 2011, grocery stores accounted for 91 percent of the $571 billion sales in the traditional food store sector, according to the U.S. Census Bureau [1]. This sector's supply chain includes suppliers of groceries, distribution centers, retail stores, and end customers. Distribution centers (DC) are particularly important in the industry to lower the costs and increase the efficiency of the entire supply chain operations. Tompkins *et al*. [2] describe the distribution center's key functions, such as receiving, transfer and put away, order picking, sortation, crossdocking, and shipping. The authors also indicate that distribution centers spend most of their resources optimizing the efficiency of *order picking* functions. As defined by [3], *order picking* is the function of retrieving from the warehouse the merchandise contained in the orders placed by retail stores. The merchandise is coded in stock keeping units (SKU's) for identification and product tracking. The literature is vast in methods used to optimize order picking, as demonstrated by the thorough reviews of the literature such as [3],[4],[5]. These methods include identifying good order-picking routes, optimizing zoning, and consolidating multiple orders in a batch [6]. These methods, in general, reduce total travel distance required to fulfill an order, and therefore they affect the warehouse's order picking time (i.e. the time to retrieve all the items in an order at the right quantities).

Distribution centers manage huge amount of data and one of the contributing factors is the proliferation of SKUs in retail stores. As explained in [2], the retail stores carry multiple versions of the same product, each version can be unique in terms of size, volume, weight, presentation, etc. The analysis of massive amount of data can assist in the detection of the ordering patterns of retail stores. Data-intensive analytical tools, such Association Rules Mining Algorithms were first proposed in [6] and [7]. In these papers, a batch of orders is created using association rules containing SKU's that are frequently ordered together, and an order picker tours the warehouse to fulfill the orders contained in this batch; this strategy is known as *batching order picking* (BOP). In our research, we restrict the scenario proposed by [2], [6], and [7] and assume that the order picker can only fulfill one order per trip; this strategy is known as *single order picking* (SOP). Our association mining algorithm also looks at all the orders placed by the retail stores and determines associations between the items that are ordered together. The items with high association are further processed and assigned to storage locations in the distribution center that are in close proximity. The information gathered from frequent association rules algorithms are used to create a layout configuration, which is tested using a discrete-event simulation model developed in WITNESS® simulation.

The paper is organized as follows. Section 2 contains a literature review. Section 3 introduces the proposed association rule algorithm, and Section 4 then presents the description of a simulation model of a virtual distribution center under analysis and the results showing the comparison of the virtual distribution center's baseline to a modified DC configuration generated by the application of the proposed association mining algorithm. In Section 5, we present concluding remarks and future work opportunities of this research.

2 Literature Review

Thomas and Meller [8] provide guidelines for designing manual, case picking warehouses. The authors provide reports that these types of warehouses have increased in recent years. The guidelines are specified in terms of decision variables such as "size and layout of the forward area", "dock door configuration", "pallet area shape", and "pallet rack height". To create the guidelines, the authors used six existing warehouses and fourteen data sets. Studies also optimize the assignment SKU's to storage locations [9].

Besides the layout configuration, order-picking has been found to be a critical warehouse operation affecting performance. Lin and Lu [10] indicate that the picking strategies, the SKU's storage policies and the picker's routing patterns are the contributing factors affecting order picking efficiency. They propose several order picking strategies, and then use simulation models to find the strategy resulting in the lowest order picking time and largest picker utilization. The Batching Order Picking strategy is discussed in Hsu *et al.* [11]. The authors proposed a genetic algorithm to form batches of multiple orders as to minimize the total distance traveled required to pick up all the orders in a batch. The algorithm, the authors claim, work for any batching structure and for any warehouse layout. Chen *et al.* [5] introduce the use of association rules mining algorithms for batching formation. A two-phase heuristic recognizes the associations between orders, and then forms the batches with orders that are highly associated. Chen *et al.* [6] present an extension of this model, which models capacity constraints of the facility by using 0-1 integer programming.

Frequently, picking lists are mostly static. However, picking lists can change dynamically in [12] to account for urgent or late orders. The authors present an Interventionist Routing Algorithm to optimize the dynamic behavior of the routes characteristic of these systems.

3 Association Rules Mining

Association rule mining, also known as *frequent item set mining*, is a well known data mining algorithm introduced by [13]). This algorithm generates efficiently the most *frequent association rules* among items in a database. The algorithm was first applied to grocery stores and supermarket data. The information obtained, such as "*2% of the transactions purchase bread and butter together*", was used to make decisions that would lead to an increase in sales.

Formulation

Agrawal *et al.* [13] formalize the model as follows. Let $I = \iota_1, \iota_2, \ldots, \iota_m$ be a binary variable and let $\mathbf{X} \in I$. An association rule is expressed as an implication,

$$\mathbf{X} \Rightarrow I_i \tag{1}$$

where $I_j \notin \mathbf{X}$; in this context, the set $\mathbf{X}$ is called the *antecedent* and I_j is called the *consequent*, and it is not a subset of $\mathbf{X}$. Three concepts related to association rules are of interest: the *support*, the *confidence* and the *lift*. To understand these concepts, let us formally define an *itemset* as a set of SKUs carried together in an order. Let us also define a *transaction* as the trip from the point in which the pallet is empty and waiting to be loaded to the point in which the order is fulfilled and unloaded at the docks. The *support* for an itemset represents how frequently an itemset is carried within a transaction. Therefore, within this context, support constitutes an overall measure of importance (relevance) for an itemset [14]. The *confidence* is the conditional probability $P\ (I_j|\mathbf{X})$; this is considered a measure of the strength of the rule. It represents how much more likely an SKU I_j is to be selected given the order $\mathbf{X}$ is being selected. The confidence leads to a most likely sequence in which items are usually selected. The *lift* is the measure of the correlation between the items in the itemset (e.g., $\mathbf{X} \cup I_j$). For instance, a lift greater than 1 implies that the fact that these items appear to be grouped together is not likely to be due to mere chance. An itemset with high support and lift greater than 1 represents a group of products which are transported together in the same transaction relatively frequently.

Algorithm

This study used the version of the *Apriori* algorithm implemented in R by [13], [15], and [16]. The use of the Apriori algorithm in this study has a dual purpose. It is used to: 1) identify groups of products that are transported together around the warehouse, the *itemsets*; and, 2) estimate the frequency with which these itemsets are transported.

The Apriori algorithm is efficient and relies on the *downward closure lemma*, which is used to discard itemsets that are not frequent. That is, it discards sets that do not meet minimum support or minimum confidence conditions [18]. In essence, let X, Y be any two itemsets. If $X \supseteq Y \Longrightarrow \mathrm{supp}(X) \geq \mathrm{supp}(Y)$. Therefore, if X is not frequent, then any subset of Y: $Y \subseteq X$; is also not frequent.

Apriori finds frequent itemsets size k or less, L_k, in a database of transactions T. It consists of three main phases: 1) find frequent itemset L_{k-1}; 2) join; and, 3) pruning.

Apriori Pseudo-code [18] :

```
Ck: a candidate itemset of size k
Lk: frequent itemset of size k
T:  database of transactions/trips

Apriori(T, ε)
L1 ← {large 1-itemsets appear in more than ε
      transactions}
 k ← 2
   while (Lk-1 ≠ ∅)
      Ck ← generate itemsets from Lk-1
      for (transactions t ∈ T)
           Ct ← subset(Ck,t) generate candidate
                                transactions size k
           for (candidates c ∈ Ct) determines
                                      frequency of
                                      c-candidates
                count[c] ← count[c]+1

      Lk ← {c: c ∈ Ck ∧ count[c] ≥ ε} Pruning
      k ← k + 1
return ∪k Lk  union of sets of frequent items
               k=1,2,... K
```

The result is an itemset L_k, which consists of those itemsets of size less than or equal to k that meet the minimum support required ε. For instance, for a set of transactions L_4, rules with 2, 3, and 4 items are possible, as shown in Table 1.

Such frequent itemsets (i.e., rules) indicate items that are frequently picked together. Knowing which items are picked together (e.g., item 1 and item 4) allows the warehouse management to store them at a short distance from each other. Storing those items close together reduce the distance that must be travelled to pick them.

Being able to identify the groups of products that are transported together allows to specify a family grouping system. Being able to estimate the frequency with which these items sets are transported allows to identify trends in joint demand for these items. Both capabilities allow for deeper analysis of local market trends and demand forecasting; however, for this application, we are more interested in improving the internal warehouse operation.

Table 1: Sets of Products

Antecedent	=>	Consequent	Support
{item 1}	=>	*{item 4}*	*0.90*
{item 2, item 5}	=>	*{item 20}*	*0.85*
{item 3, item 14, item 5}	=>	*{item 60}*	*0.75*

Ideally, we are searching for implications or itemsets that have a large support, large confidence and lift greater than 1. In certain cases, an itemset may show great confidence, but low support. This implies that the the information conveyed by the rule does not become available very frequently.

Dataset description

The dataset under study contains approximately 123,000 records. The dataset describes the orders that retail stores send to the distribution center. The data fields include among other things the SKU per order, the order quantity per SKU, Product Description, Aisle # and Position (location of SKU in the Distribution Center), Case Dimensions, Case Weight, and the Current Inventory On Hand per SKU.

4 Simulation Model and Results

The method presented in Section 3 is evaluated with computer simulation. Data-intensive analysis results on a plan to group into families the products that are ordered together and allocate the storage position of these products with close proximity in the distribution center in order to improve the distribution center's operations; i.e. reduce the total order picking time; the reader should note that the allocation of products to storage positions is beyond the scope of this paper, so this paper ommits the details. The simulation model consists of evaluating the performance of the algorithms and comparing the distribution center's layout configuration to other layouts generated by other algorithms. The simulations are described below.

4.1 System Description and Simulation Model

A virtual distribution center is assumed for evaluating the algorithm presented in Section 3. The distribution center's configuration is as follows. Orders are received at time 0 by more than 500 unique retail stores of one company's food retail chain. There are 624 orders in this study, and each order contains 4-350 unique items or SKU's. The SKU's have fixed storage positions assigned. There are 6480 unique positions to store more than 5,000 different SKU's. The warehouse has 24 aisles; there are two bays per aisle; and there are 135 positions per bay.

The order picking process consists of a human operator (order picker) operating a powered vehicle. The operator receives one order at a time. The operator picks SKU's according to a First-In First Out (FIFO) sequence prescribed by the warehouse management system. An important criterion in the route formation is to sequence the picking jobs according to weight. The heaviest goods will be scheduled near the beginning of the picking route so that they can be used to form the base of the pallet, followed by less heavy items so that the fragile items set in the top of the pallet. The operator arrives at the SKU's location and picks the number of required items. The operator then follows the route to pick up all the SKU's contained in the order. The order picking time is correlated to the number of items. Once the order is picked, the load is palletized and send to the docks, where the order will be loaded into a truck and transported to the customer (retail store).

A full-scale simulation model of the virtual distribution center was created in WITNESS® Simulation vs. 13 with the capacity to read up to 1,000 orders, use 40 vehicles, read 6,480 different item locations, and coordinate 40 wrapping stations to give service as needed to the selectors. The model uses up to five powered vehicles operating between the facility concurrently. The flow in the warehouse is set to unidirectional mode. Similarly, the number of vehicles per aisle is limited to one. The implication is that, any other vehicle that needs to access the aisle will need to wait until the vehicle currently occupying the aisle finishes fetching the items from that aisle. These rules are implemented in this study to simplify the complexity of the simulation model; yet, the simulation model allows to study vehicle congestion patterns caused by the allocation of SKU's to the storage positions in the distribution center.

To test the performance, the DC layout created with the association rules algorithm is compared to a baseline configuration. The description of the layouts is as follows.

- *Baseline configuration:* The virtual distribution center's layout is modeled after an actual distribution center. The location of the SKU's is assigned based on ad hoc rules derived from years of experience operating this facility.
- *Modified configuration:* The algorithm was used to reveal associations among the SKU's contained in the 624 orders which are the first set of items that are fetched frequently. As specified in Section 3, this information is obtained from the *support*. Likewise, the importance of the associations found are assessed by the *lift*. In this study, support = 0.75, confidence = 0.75. Setting the parameters to these values also helped reducing the computational time of the algorithm. A heuristic was then used to assign a fixed location to the SKUs contained in the output rules.

4.2 Results

The algorithm was applied to 20 orders randomly selected from the population of 624 orders. The reason why the analysis was constrained is to reduce the computational time required to run the data-intensive analysis.

Table 2: Sets of Products

Antecedent	=>	Consequent	Support	Confidence	Lift
{SKU 1}	=>	*{SKU 2}*	*0.9048*	*1.0000*	*1.0000*
{SKU 2}	=>	*{SKU 1}*	*0.9048*	*0.9047*	*1.0000*
{SKU 3}	=>	*{SKU 2}*	*0.9048*	*1.0000*	*1.0000*
{SKU 2}	=>	*{SKU 3}*	*0.9048*	*0.9047*	*1.0000*
{SKU 1}	=>	*{SKU 3}*	*0.8095*	*0.8947*	*0.9889*

Frequent itemset mining is used to identify the sets of items that could be rearranged inside the warehouse in such a way that the overall performance of the product fetching process can be made more efficient. Table 2 shows, for illustration purposes, a selected sample rule generated by the algorithm. The first rule (Row 1) indicates that 90.48% of the orders contain SKU 1 and SKU 2. These SKU's were located on separate aisles in the baseline configuration. Clearly, relocating these two products can reduce the order picking time. The resulting storage locations, for example, for these SKU's are shown in Table 2. The process was applied to all the SKU's in the 20 orders under analysis to find improved positions for these SKU's.

The simulation results are shown in Table 3 for the 20 randomly selected orders. The maximum number of SKU's in the orders sampled are 360, and the aggregated total number of SKU's is 2923. The results of the simulations for the 20 randomly selected orders is shown in Table 2. In the Baseline Layout, the Total Order Picking Time for 20 orders is 10.4 Hours. In the Modified Layout, the Total Order Picking Time decreased to 9.1 hours assuming that the conditions for the distribution remained the same. This represents an improvement of 12%, which is significant for these types of environments.

Table 3: Sample Assignment of Three SKU's to Fixed Storage Locations of the Distribution Center

SKU	Weight (lbs)	Original Aisle Name	Original Position	New Aisle Name	New Position
SKU 1	*9*	*A38*	*396*	*A40*	*3*
SKU 2	*5*	*A40*	*901*	*A40*	*6*
SKU 3	*5*	*A39*	*284*	*A40*	*2*

Table 4: Simulation Results for 20 Randomly Selected Orders

Results for a Sample Order	Number of SKU's	Total Order Picking Time (Hours)
Baseline Layout	2923	10.4
Modified Layout	2923	9.1

5 Conclusions

In this research, data intensive analysis and simulation was used to improve the performance of a distribution center. By using a simulation model of a virtual DC of a merchandise retail company, we evaluated the layout configuration generated by a association mining algorithm that groups items ordered together into families. These families were assigned to storage positions in close proximity to eliminate travel time. The simulation model evaluated this layout configuration, and compared the results to a baseline configuration. The simulation results show an improvement of the total order picking time for 20 orders.

The main challenge was the time requirements associated for taking the massive amount of available data and extracting useful information from it. Further research include the testing of larger simulations (i.e., more orders). Due to the heuristic nature of the solution presented in this study, it is possible that the use or a association-based storage system design will result in a more drastic improvement than the one shown in this study.

These results trigger important implications for managers at operational and tactical levels. Some of these implications include: (1) Warehouse managers can increase the efficiency of the DC by increasing the number of orders fulfilled in any given period. (2) Order accuracy should be higher. By having items within the DC grouped based on the support and lift of the itemset, the probability of having the correct items in the order is higher. Minimizing returns to the DC and having more correct orders reflect in more processing of arriving orders and higher productivity of the DC. (3) Minimizing returns implies that retailers are most likely receiving what they asked for and consequently retailers will have the products their customers want. (4) Better DC layouts will reflect in less travelling time within the DC, better use of resources, and lower carbon footprint. Better utilization of available resources are clearly a win for all partners in the process.

References

1. U.S. Census Bureau, Annual Retail Trade Report, http://www.census.gov/retail
2. Tompkins, J.A., White, J.A., Bozer, Y.A., Tanchoco, J.M.A.: Facilities Planning. 4th ed. John Wiley & Sons, New Jersey, 2010
3. van den Berg, J.P.: A Literature Survey on Planning and Control of Warehousing Systems. IIE Transactions. 31(8), 751--762 (1999)
4. De Koster, R., Le-Duc, T., Roodbergen, K.J.: Design and Control of Warehouse Order Picking: A Literature Review. European Journal of Operational Research. 182(2), 481--501 (2007)
5. Gu, J., Goetschalckx, M., McGinnis, L.F.: Research on Warehouse Operation: A Comprehensive Review. European Journal of Operations Research. 177, 1--21 (2007)
6. Chen, M.-C., Huang, C.-L., Chen, K.-Y., Wu, H.-P.: Aggregation of Orders in Distribution Centers Using Data Mining. Expert Systems with Applications. 28(3), 453--460 (2005)
7. Chen, M.-C., Wu, H.-P.: An Association-Based Clustering Approach to Order Batching Considering Customer Demand Data. Omega. 33, 333--343 (2005)
8. Thomas, L.M., Meller, R.D.: Developing Design Guidelines for a Case-Picking Warehouse. International Journal of Production Economics. 170, 741--762 (2015)
9. Brynzér, H., Johansson, M.I.: Storage Location Assignment: Using the Product Structure to Reduce Order Picking Times. International Journal of Production Economics. 46-47, 595--603 (1996)
10. Lin, C.-H., Lu, I-Y: The Procedure of Determining the Order Picking Strategies in Distribution Center. International Journal of Production Economics. 60-61, 301--307 (1999)
11. Hsu, C.-M., Chen, K.-Y., Chen, M.-C.: Batching Orders in Warehouses by Minimizing Travel Distance with Genetic Algorithms. Computers in Industry. 56, 169--178 (2005)
12. Lu, W., McFarlane, D., Giannikas, V. Zhang, Q.: An Algorithm for Dynamic Order-Picking in Warehouse Operations. European Journal of Operational Research. 248, 107--122 (2016)
13. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In *ACM SIGMOD Record*, Vol. 22, No. 2, pp. 207-216) (1993).
14. Han, J., Kamber, M., Pei, J.: .Data mining: concepts and techniques: concepts and techniques. Elsevier (2011).
15. Borgelt, C.: Efficient implementations of apriori and eclat. In *FIMI'03: Proceedings of the IEEE ICDM workshop on frequent itemset mining implementations* (2003).
16. R Core Team: R: A language and environment for statistical computing. *R Foundation for Statistical Computing*, Vienna, Austria, (2015).
17. Omiecinski, E. R.: Alternative Interest Measures for Mining Associations in Databases. Knowledge and Data Engineering, IEEE Transactions on, 15(1), 57-69 (2003).
18. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In Proc. 20th int. conf. very large data bases, VLDB , Vol. 1215, pp. 487-499 (1994).