

Warehousing Efficiency in a Small Warehouse

Veronique Limere
Ghent University
Zwijnaarde (Gent), Belgium
veronique.limere@ugent.be

Melih Celik
Georgia Institute of Technology
Atlanta, GA. USA
apradhan@gatech.edu

Aditya Pradhan
Georgia Institute of Technology
Atlanta, GA. USA
melihcelik@gatech.edu

Mallory Soldner
Georgia Institute of Technology
Atlanta, GA. USA
msoldner@gatech.edu

Abstract— Small warehouses generally have different needs than large warehouses. They usually do not have warehouse management systems that are data intensive and involve high capital investment. Operational procedures are more nebulous and management control is less rigid. Because of the difference in operational approach, different measures are needed in order to enhance productivity. This paper describes the results and insights gained from a study of the inventory control and warehouse operations at an industrial distributor of maintenance and repair items. The accuracy of the inventory and the efficiency of order picking are studied and appropriate measures are proposed in order to improve operations. Improvements are in the areas of process organization, inventory accuracy, inventory control, and order picking. Implementations and results are reported. Major improvements include lowering inventory levels and more efficient order picking.

Keywords—Inventory control, order picking, order batching, heuristic, small warehouse.

I. INTRODUCTION

SMALL warehouses are usually quite distinct from large warehouses in a number of ways. First of all, typically in smaller warehouses stock-keeping unit (sku) density is quite high. Capital investment is quite low. Small warehouses usually do not invest in costly warehouse management systems, and automation is also lacking or limited. These differences necessitate different needs for the internal operations management. To be efficient, processes in small warehouses need to be carefully structured in order to meet productivity and accuracy goals.

In this paper we describe the results and insights acquired during a case study at an SME (Small and Medium Enterprises) warehouse. In order to respect client confidentiality and by request, the company's identity will not be revealed. The warehouse belongs to a small industrial distributor of maintenance and repair items. The parts they store and distribute are drill bits, nuts and bolts, hand cleaner, etc. They receive rather small orders, around 50 orders a day and 8-10 lines per order. Their warehouse stores about 25,000 different part numbers and is quite small—about 550 square

meters. At the request of the client, their operations were examined and recommendations for improvement were made. The main problems indicated by the client were inventory accuracy and an inadequate fill rate.

Preliminary data analysis revealed that the warehouse has improvement opportunities in the following areas:

- Inventory accuracy
- Inventory control
- Order picking

Improvement recommendations in each of the areas are described in the following sections.

II. INVENTORY ACCURACY

Inventory accuracy was indicated by the client as an important focus area. Current accuracy levels were unsatisfactory at around 90 percent. Inaccurate inventory levels negatively influence the fill rate and lead to double work and operator frustrations as well as lower customer satisfaction and higher costs due to backorders.

The daily process flow used by the client is represented in Figure. 1. Orders are entered by sales representatives a day in advance. In the morning, orders are printed and orders that are completely in stock – according to the book inventory levels – are picked. Between 9:30 and 11:00 AM inbound merchandise arrives and is received in the system. Order picking of complete orders is continued based on the updated book inventory levels. Finally orders are checked, packaged and shipped. As a consequence of low accuracy levels, order picking is often interrupted because of unexpected shortages.

A survey of 410 manufacturing companies indicates that companies using cycle counting methods achieve higher levels of inventory record accuracy, compared to companies that use physical inventory to measure inventory record accuracy [1].

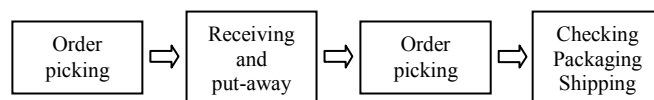


Fig. 1 Process flow

Therefore, cycle counting is proposed in order to reduce the frequency with which operators are confronted with insufficient stock levels in the middle of the picking process. Cycle counting is defined as ‘any process that uses regularly scheduled counts but does not count the entire facility’s inventory in a single event’ [2].

In order to reduce interruptions of order picking operations, a cycle counting program was developed. The basic premise behind the plan is that items for which ensuring inventory accuracy is most important are those that are to be picked today and are those that are close to their reorder points. The items to be counted are therefore determined by comparing inventory levels with reorder points. If an item is within a given percentage, deemed the ‘count tolerance’, the item is chosen to be counted.

$$OHI \leq (1 + \alpha) \times ROP \tag{1}$$

Where,

- OHI = on-hand inventory
- α = count tolerance
- ROP = reorder-point.

This inventory counting scheme is a form of opportunity-based cycle counting, which means that counting is performed at particular key events in the process [3], in this case when the part is ordered. The plan follows a “count late, count less, count often” principle. This allows the worker to count the minimum number of items while preventing backorders by counting at a point close to the reorder point.

In order to easily implement this procedure, an Excel interface with embedded Visual Basic macros has been developed in order to assist with inventory counting. A screen shot of the interface, for a standard day, can be seen in Fig. 2.

	A	B	C	D	E	F	G	H
1	Count Tolerance	0.25						
2								
3	Total Order Lines	144						
4	Unique Items	103						
5	Items to be Counted	20						
6								
7	Clear Daily Data							
8								
9	Determine Items to be Counted							
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								

ITEM	LOCATION	ONHAND	ONORDER
55574	20A05	37	2000
21564	70000	0	0
95929	19B05	1	0
95105	27B03	1	0
14584	05A05	150	2000
14592	05A05	95	540
800041	06C03	225	0
95836	06C05	0	0
95929	19B05	1	0
50744	19B04	12	288
95929	19B05	1	0
50744	19B04	12	288
88040	25A08	0	200
36344	20A05	20	0
53298	22B09	0	1
12414	06D02	21	400
14804	06A06	4	2500
50744	19B04	12	288
50742	19B04	97	0
53262	22B03	46	0

Fig. 2 Cycle count plan – interface

Order lines and inventory files are inputs. Reorder points for items are preloaded (see Section III for methodology related to setting reorder points), but can be easily updated to accommodate desired changes or additional items. Two buttons are used to execute the macros. The “Clear Daily Data” button clears the orders and inventory from the workbook in the event that the previous day’s orders and inventories are already in the file. Once the desired count tolerance has been set, the “Determine Items to be Counted” button selects a subset of the day’s items to be counted. This keeps the number of items to be counted on each day within reasonable limits as shown in Fig 2.

A day’s orders are tested for the sensitivity of the number of items whose on-hand inventory is close to the count tolerance. Results are graphically represented in Fig. 3. A reasonable count tolerance should be determined based on the available time and the desired control.

Ideally one should be able to count and resolve discrepancies in the system with no other processing going on during the count process. In order to support this, the process flow is adjusted (Fig. 4). After printing the orders of the day, a maximum time should be determined, based on the workload of the day. This time is the maximum time that should be spent doing cycle counting at the beginning of the day in order to ensure shipping deadlines are met. The sub-processes framed by a dotted line only take place in certain cases. Order picking is only done before receiving and put-away if time is left after cycle counting, or if the workload for the day is considerably high so that postponing is not recommended. In that case, cycle counting will be postponed until the end of the day or the next day. On a slow day, all counting can be done before receiving and put-away. This has an additional advantage that order picking can be done at once after the daily update of the inventory levels.

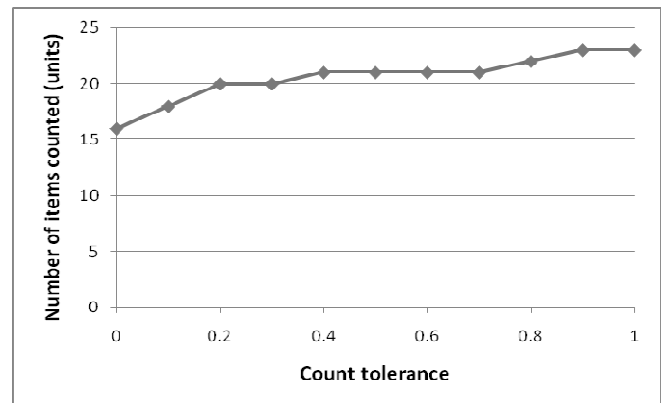


Fig. 3 Sensitivity of the count tolerance

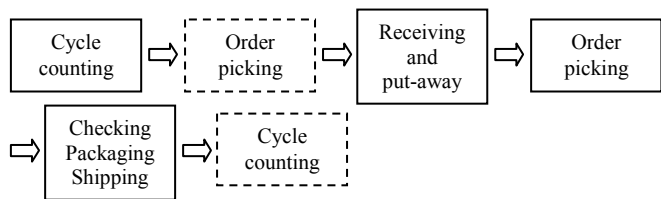


Fig. 4 Revised process flow

III. INVENTORY CONTROL

Once a reasonable level of inventory accuracy is achieved, the focus can be shifted to inventory control for improvement opportunities. High backorder levels were also indicated as a problem by the client. Although better accuracy decreases the number of backorders, better inventory levels based on historical data help ensure the fill rate is improved.

Inventory analysis was used to quantify and compare the amount of expected and actual inventory levels of each stock keeping unit. Once computed, these levels are translated into their associated holding costs. A base stock or order-up-to model [4] to theoretically approximate the real inventory was chosen as the most suitable statistical inventory model. The model was adapted to account for box quantities, since client orders are always in a multiple of box quantities. For the sake of the analysis, demand was assumed to follow a normal distribution. In terms of inventory holding costs, 25% of the item cost was represented to be the holding cost for a particular item in consideration. This comes from a maintenance cost of 20% and a 5% interest rate to quantify the time-value of money. The inventory model assumes an 85% fill rate, i.e. the current fill rate as indicated by the client. Shortage costs may therefore be ignored because the model will be comparable with reality. The final analysis revealed that a significant amount of excess inventory is held. Table I shows an aggregate comparison between the calculated (appropriate) inventory level and the real inventory level.

Fig. 5 represents the actual holding versus the theoretical holding costs for each stock keeping unit. The x-axis is ordered for a decreasing theoretical inventory level. Because the actual holding cost is a snapshot at one point in time, and because a conservative approach is preferred, the actual holding costs are compared with the maximum theoretical holding cost under an assumption of constant demand, i.e. double the expected theoretical holding cost. This was done because historical inventory levels were not available due to lack of data – a common problem in small warehouses. The low fill rate and significant amount of excess inventory is an indicator of inventory mismanagement at the warehouse implying significant room for improvement.

The analysis reveals a significant amount of excess inventory. Opportunity for improvement clearly exists, but a careful interpretation is recommended. First, data reliability is very important to support the analysis. It should be checked thoroughly to ensure that all of the inputs are correct. Moreover, the theoretical cost savings may be difficult to

reach because of practical considerations. Ordering is done per vendor and batching can be done to avoid additional transport costs. Moreover, some items are extremely in excess, and gradually reducing the stock will take a long time. It is better to liquidate part of the stock of these items directly. This may mean that only part of the savings can be realized.

TABLE I
INVENTORY LEVEL & HOLDING COSTS – ACTUAL VERSUS THEORETICAL

	Theoretical	Actual	Excess
Total inventory (units)	734.654	1.229.872	40%
Total holding costs (\$)	675.748	1.286.775	47%

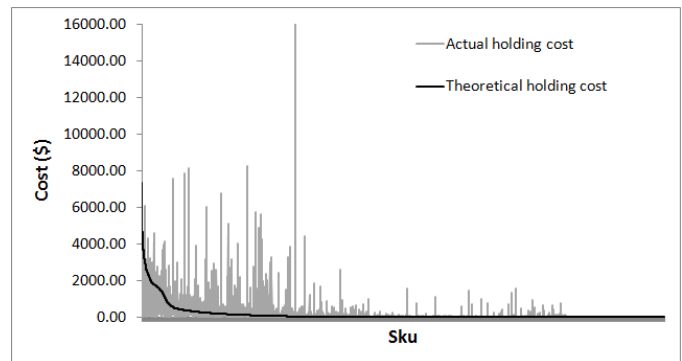


Fig. 5 Actual versus theoretical holding cost

The primary goal of this analysis was to give more insight into the problem areas as well as the extent of the problems. In a first phase, it is recommended that the part database should be checked and as many measures as possible should be taken immediately.

- Items with on-hand inventory levels that are more than the double of the calculated inventory levels should be checked and possibly part of the stock should be liquidated.
- Items with zero average demand and a positive on-hand inventory take up unnecessary space, money, and maintenance. It is unprofitable to keep these items in stock as they are not demanded anymore by the customer.
- Items with a positive on-order amount and an on-hand inventory level which is higher than the reorder point are ordered too early. On the other hand, items without an on-order amount and an on-hand inventory level which is lower than the reorder point are ordered too late.

It is important to focus primarily on the stock keeping units that have the highest calculated inventory holding costs. These are the ones that either will have a high inventory level and will take up a lot of space in the warehouse, and/or will have a high item cost, and will as such have a large influence on profits.

In the future, it is important to keep the real inventory levels

within a certain range of the appropriate level and this can be done by ordering through a reorder point policy, as determined by the order-up-to model. Appropriate reorder points for each stock keeping unit are defined.

IV. ORDER PICKING – BATCHING AND ROUTING

If inventory is under control, order picking can be executed effectively. Order picking is the most important process in most warehouses because it consumes the most labor and it determines the level of service experienced by the downstream customers [5]. Order picking typically accounts for about 55% of warehouse operating costs and traveling comprises the greatest part of the expense of order picking [6]. Because small warehouses do not have costly warehouse management systems, picking routes are usually not generated by the system. This is the case at the client’s warehouse, and so a batching algorithm is proposed.

A. Overview

When picking orders, two options are possible: namely single and batched. Single order picking, in which an order-picker picks items corresponding to a single order in a single trip, is especially useful when: (i) the sizes of the orders are large; (ii) the time between two orders is large enough so that waiting for the orders to be accumulated increases the order fulfillment time substantially; or (iii) when the capacities of the order-pickers do not allow picking multiple orders in a single trip. On the other hand, when the sizes of the orders are small, and when the capacities of the order-pickers allow picking multiple orders in a single trip, order batching can be applied. There are two main methods for order batching. Proximity batching, in which assignment of orders to the batches are made based on the proximities of their storage locations, is the first of these. The main issue in this method is to determine a proximity measure for the orders. The second method is time window batching, which forms batches from the orders that arrive in the same time interval, which is called a time window. The orders are then picked in the same trip in the following stages.

In the case of the warehouse for which the study has been conducted, order batching has two advantages. Firstly, due to the size of the warehouse, usually several orders lie in high proximity to each other each day. This makes single order picking unreasonable as picking the orders in batches saves significant travel time. Secondly, the sizes of most orders are small so that capacity of the picking cart does not hinder batch picking of the orders. The objective of the order batching analysis conducted throughout this study is to define a standard method for batching orders in a small warehouse.

There is a vast amount of literature regarding order batching, but most of these studies tackle the order batching problem using a theoretical approach, and most ignore the applicability aspect of the algorithms. For instance, Gademann and Van De Velde [7] discuss proximity batching in a warehouse with a single input/output station, and show that

the order batching problem is polynomially solvable when the maximum number of batches is two, but the problem is NP-complete when the number of orders in a batch is allowed to be higher. More relevant to our study is the work of Won and Olafsson [8], who consider the order batching and order-picking problems together. They assume a warehouse where end-of-aisle order-picking is applied, the distance metric is Tchebyshev. They propose an integer programming formulation for the joint batching and routing problem and propose two heuristics that solve the problem sequentially. Ideas are borrowed from the formulations of the bin packing problem for order batching and the travelling salesman problem for the order-picking parts, which makes the heuristic procedure hard to apply. Gray et al. [9] consider the batching problem within a wider context, and present a case study where item location, zoning, picker routing, pick list generation, and order batching problems are considered together for a large-sized warehouse. They propose a hierarchical solution process, which is case-specific so hardly applicable to other warehouses. Petersen and Aase [10] tackle the storage assignment and batching problems together, and test the performances of different heuristic procedures to minimize the travel distance.

The warehouse does not apply any standardized means of order batching and picker routing. Thus, in this study, we propose an automated angle-based proximity batching heuristic that takes as an input the contents and locations of a day’s orders and then provides as an output batches of orders to be picked together. In addition, we use a serpentine heuristic to decide upon the order in which individual pick lines should be picked within a batch. The standard routing scheme invoked by the line-ordering heuristic draws a framework in which the picks can be sequenced in an efficient manner and helps measure the performance of the proposed batching procedure. Simplicity was a goal along with effectiveness.

Model and Analysis

The algorithm uses a central point in the warehouse defined along with a bisecting line that allows assigning each warehouse location an angle that is unique. While locations along the same radial axis may have the same angle, this does not cause significant deviation in performance of the algorithm.

The batching procedure aims to batch the orders with as much proximity to each other as possible. Here, proximity of the orders is measured by the angles they “sweep” with respect to a central point within the warehouse. Fig. 6 shows an example, in which two orders are batched together. It is obvious that the black order “spans” a larger portion of the warehouse and, further, that the span of the grey order is contained within that of the black one. So, instead of picking these orders separately, batching them together saves walking time. The span of an order (i.e., the angles in the warehouse that it sweeps) is identified by recognizing that the minimum

and maximum angles of the order.

In order to measure the performance of the given procedure and to propose a standardized pick-sequencing heuristic, an ordering and routing scheme is developed. Within a batch, pick lines are sequenced in the order which they appear in a serpentine path starting from the shipping area, which is in the lower-left quadrant of the warehouse. The picker is given the liberty to use shortcuts and detours whenever possible. The effective result, as shown in Fig. 7, is that the lighter segment of the serpentine path is only traversed once (when the black and grey orders are batched together) rather than twice (once for the black order and once for the grey order). The lighter segment is thus the effective reduction in picking distance as considered by the upcoming analysis.

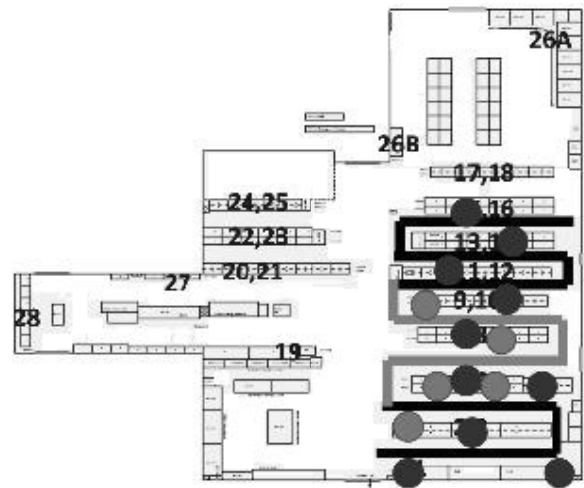
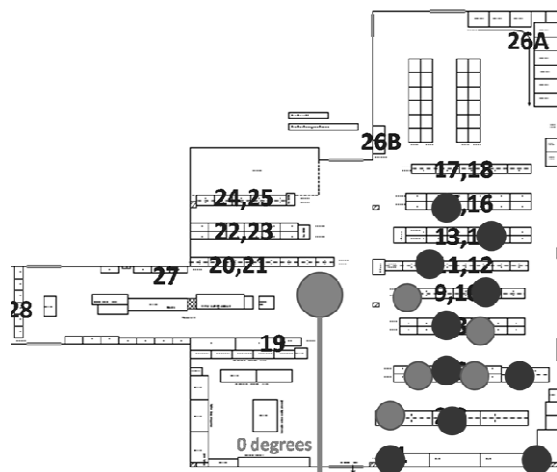
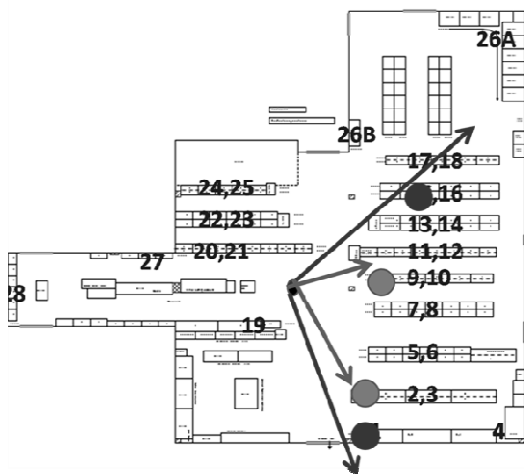


Fig. 7 The distance-savings by batching the black and grey orders



(a) Two orders (grey and black) that need to be picked



(b) The span of both orders

Fig. 6 An example for how two orders can be batched together

Thus, the batching and routing procedures are decomposed into a “cluster-first route-second” approach in order to facilitate a simple but effective algorithm for minimizing distance travelled.

Fig. 8 summarizes the batching algorithm. The routing algorithm consists of determining a sequence of items, sorting the items in each batch according to that sequence, determining a serpentine path which completely travels each aisle with an item to pick in it, and having the picker make use of the shortcuts whenever possible.

Results

The performance of the batching procedure has been tested using a Java code for routing, using actual demand data from 2009. Three sample days have been selected from the past year’s data from three categories of days: busy, average and light. In the distance calculations used to approximate the savings of the batching algorithm in the company’s operations (when compared to picking each order separately), we assume that given a pick sequence, the picker will take the shortest distance between two consecutive orders. This allows deviation from strictly travelling on the serpentine path, but it does not capture shortcuts that are possible by swapping the order of two pick lines. However, due to the small size of the warehouse and the density of the picks, the approximation is not a far deviation from the actual savings assuming optimal route planning.

When batching orders, there is a tradeoff between savings in distance travelled vs. savings due to making the picking more complex. For example, batched orders must be sorted either at the time of the pick, into a pick cart with multiple baskets, or after all lines have been picked. For this reason, sensitivity testing is done by varying the maximum limits in the algorithm on the number of orders per batch and the number of lines per batch. For each day, the maximum number of orders in a batch is varied between 1 and 10, and the maximum number of pick lines in a batch is varied

between 10 and 100 in increments of 10. In both cases, the law of diminishing returns applies. In other words, as the maximum number of orders or maximum number of pick lines in a batch is increased, the marginal benefit from increasing the capacity of the batch by a single unit decreases.

Algorithm 1 – Batching Algorithm

```

minj = maxj = +∞ ∀j
for each order j do
  for each line i in j do
    if anglei < minj then
      minj = anglei
    end if
    if anglei > maxj then
      maxj = anglei
    end if
  end for
  if order j has two items then
    if maxj - minj ≤ 180 then
      spanj = maxj - minj
    else
      spanj = 360 - maxj + minj
      min'j = maxj - 360
      maxj = minj
      minj = min'j
    end if
  else
    Find the maximum angle difference between two
    angles on the sorted angle list. Assume this happens
    between items k and l, where k and l represent the orders
    on the sorted list.
    if 180 ≤ anglel ≤ 360 then
      spanj = 360 - anglel + anglek
      minj = anglel - 360
      maxj = anglek
    else
      minj = anglek
      maxj = anglel
    end if
  end if
end for
while there exists an order not in a batch do
  Start a new batch, order j with the maximum spank will be
  the first order in the batch.
  while capacity of the batch not exceeded and there exist
  orders not checked do
    Start with the least sized order i. If mini and maxi
    are within minj and maxj, add i to the batch if
    capacity allows. Repeat for other items in increasing
    order size.
  end while
end while

```

Fig. 8 Batching Algorithm

It is observed that depending on the size of the batch in terms of orders, the average improvement is between 27.43% and 67.80%. Similarly, when the limit is on the number of pick lines, the average improvement is between 31.05% and 73.36%. It should be noted here that these values are actually upper bounds on the total savings when compared to the current procedure as the analysis assumes that no batching is currently applied. Nevertheless, it provides a measure of how efficiently the proposed procedure performs. There are also

savings that cannot be quantified: the pickers no longer have to spend time on how they should batch the orders. Furthermore, the pick sequences are determined by the program, so this will provide additional savings from the time required by the pickers to decide on what route they should follow while picking. Fig. 9 and Fig. 10 summarize the results.

Varying the input parameters of the algorithm independently shows that the improvements gained were asymptotic to approximately 70% of the total distance that would be travelled if each order was picked individually. This improvement is irrespective of the number of orders in a day. The user must balance the need to achieve improvements (increase number of orders per batch and number of lines per batch) with the need to reduce picking complexity (reduce number of orders per batch and number of lines per batch). This may be a user-defined trade-off, though the data suggests that a max orders value of 5 and a max order lines value of 50 should achieve significant improvement for a reasonable increase in picking complexity. It must be noted that due to the nature of the batching algorithm, it only works well for warehouses that have a small radial distance as defined from the central point.

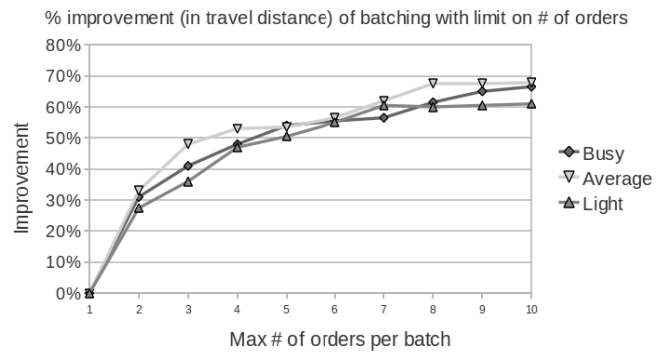


Fig. 9 Summary of the results when the capacity is in terms of maximum number of orders in a batch

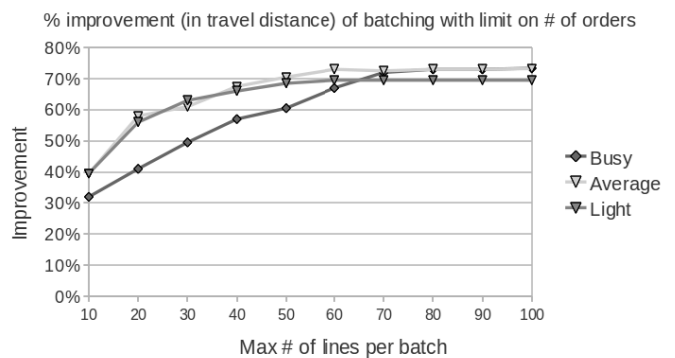


Fig. 10 Summary of the results when the capacity is in terms of maximum number of pick lines in a batch

This part of the study demonstrates that even a simple batching and routing algorithm can produce gains in small warehouses. While the absolute value in minutes saved per

day may be low due to the smaller scale of daily pick minutes, when considered as part of a holistic warehouse improvement initiative the time gained from reduced picking time can be important. Larger warehouses lacking substantial IT investment could also implement this algorithm in order to gain significant reductions in picking distances at minimal cost, especially if order picks tend to be concentrated in certain regions of the warehouse.

V. CONCLUSION

Small warehouses have to be managed differently than large warehouses. Control is less rigid and efficiency depends more on operator experience and goodwill. Extra tools and models can help to effectively manage operations and aid in keeping small warehouses competitive with larger players in the market.

The paper describes the actions undertaken at an SME company and the associated results and insights acquired. The proposed measures do not involve costly investments but allow acting smart without a sophisticated warehouse management system

ACKNOWLEDGMENT

We would like to thank M. Bellamy, S. Borin, and S. Krishnan for their collaboration in the study, and J. Bartholdi

and P. Viehweg for their guidance.

REFERENCES

- [1] R. B. Brooks and L. W. Wilson, *Inventory Record Accuracy: Unleashing the Power of Cycle Counting (The Oliver Wight Companies)*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2007.
- [2] D. J. Piasecki, *Inventory Accuracy: People, Processes, & Technology*. Ops Publishing, 2003, Glossary.
- [3] M. Rossetti, "Inventory ARCHIVES," *Industrial Engineer*, Jun2008, Vol. 40 Issue 6, pp. 32-37.
- [4] G. Cachon, and C. Terwiesch, *Matching Supply with Demand: An Intro to Operations Management*. McGraw-Hill/Irwin, 2005, pp.242-279.
- [5] E. H. Frazelle, *World-Class Warehousing*. Logistics Resources International, Atlanta, GA, 1996.
- [6] J. J. Bartholdi, and S. T. Hackman, *Warehouse & Distribution Science*. 2010, www.warehouse-science.com.
- [7] N. Gademann, and S. Van De Velde, "Order batching to minimize total travel time in a parallel-aisle warehouse," *IIE Transactions*, 37, 2005, pp. 63-75.
- [8] J. Won, and S. Olafsson, "Joint order batching and order picking in warehouse operations," *International Journal of Production Research*, 43-7, 2005, pp. 1427-1442.
- [9] A. E. Gray, U. S. Karmakar, and A. Seidmann, "Design and operation of an order consolidation warehouse: models and application," *European Journal of Operational Research*, 58, 1992, pp. 14-36.
- [10] C. G. Petersen, and G. Aase, "A comparison of picking, storage and routing policies in manual order picking," *International Journal of Production Economics*, 92, 2004, pp. 11-19.