Multi-Floor Warehouse Planning

Samuel Krivánek, Hana Rudová, Václav Sobotka

Faculty of Informatics, Masaryk University, Brno, Czech Republic
hanka@fi.muni.cz

Abstract

We will introduce complex real-world warehouse planning problem we deal with in cooperation with the company Notino. Planning of tens of thousands of orders containing several products is handled daily in a multi-floor warehouse. We aim to solve the order-picking problem that involves organizing the orders to be picked and the manual pickers collecting particular orders. We will discuss components to solve particular tasks of the order-picking problem and its solution approaches, which are being implemented together with the warehouse simulator. We will concentrate on the application of the heuristic methods that are applied to handle the multi-floor design of the warehouse efficiently. We will summarize the ongoing results of our current work on real-life data with orders covering a half year of processing.

Introduction

Efficient warehouse operations are crucial in any supply chain (Gu, Goetschalckx, and McGinnis 2007). They are represented by receiving, storage manipulation, order picking, and shipping. Our concentration lies in order picking (Pardo et al. 2024; van Gils et al. 2018b; Vanheusden et al. 2023), becoming the most expensive warehouse operation due to extensive labor or capital demands. More precisely, our study concentrates on batch picking, in which a picker collects items for several orders on each tour (Casella et al. 2023). Moreover, we consider the warehouse with multiple floors, representing a less frequent problem in the literature (Pardo et al. 2024). It can be considered a progressive zone picking problem (Yu and de Koster 2009), where the batch of orders is passed from one zone/floor to the next. Again, it is considered to be a problem receiving little research attention in combination with other planning problems (van Gils et al. 2018b). Overall, the paper introduces this complex warehouse planning problem with its components in a real-world setting. We demonstrate the application of several heuristics for handling the multi-floor component of this problem. Simulation results demonstrate the performance of particular heuristics, providing grounds for further research and demonstrating the potential of this problem to the planning community.

This work is studied in cooperation with the Notino company, an online retailer that offers a wide range of beauty, cosmetics, and fragrance products. Founded in 2004, it is a large company with 2,500 employees, with the fiscal turnover being 1,035 million EUR in 2022. Their annual increase of 32 % promises challenging problems to solve now as well as in the future. They have 50,000 orders with 3.5 products on average per day. In the last Black Friday, about 270,000 orders were accomplished. This work is based on their production data from a half year of processing with 20,000 to 160,000 daily orders in the Czech warehouse.

Figure 1: Single floor in our warehouse with the middle two-way cross-aisle and one-way aisle with the storage boxes.

Problem Description

The company has two warehouses, one for the long-term storage of products and the second for the storage of products for the next few days. We will describe the planning processes in the short-term warehouse, where there are eight floors with a rectangular layout of low-level storage racks intended for manual product picking (see Figure 1). The warehouse is used in the business-to-customer setting, so a case storage system is used to store products, more precisely stock keeping units (SKUs), referring to uniquely identified stock items (Van Kampen, Akkerman, and Pieter van Donk 2012). A mixed shelves strategy is applied, i.e., different products may be in one storage box, and one product may be in several storage boxes (Weidinger 2018).

We consider the order-picking problem, where the input is represented by a stream of orders coming in real time. Each order has its limit time and contains several products with a given volume. A sort-after-pick process is realized such that all SKUs for each order are collected in one (cardboard) pick box, and several unsorted orders are collected into each pick box. A manual picker-to-part system is considered, which means that manual pickers walk with the picking cart through the pick locations of orders. The picking cart contains up to eight pick boxes. SKUs for orders in the pick box may need collection from several floors. The conveyor belt sequentially passes through each floor, moving pick boxes across floors. Pick boxes must visit floors containing at least one reserved SKU, bypassing floors without reserved SKUs. The base objective of our problem is to minimize the number of late orders (Chen et al. 2015; Scholz, Schubert, and Wäscher 2017) and the number of pickers used throughout all floors.

The solver of the order-picking problem can be decomposed into several components. First, a *batch selection* (task 1) is realized to select a set of orders for processing. Next, an *order batching* into pick boxes (task 2) is processed, i.e., (most of) the orders from the batch are assigned to pick boxes. Note that orders from the current batch are taken together with orders from the previous batch, resulting from several pick boxes with sufficient volume of their products. The next step is the *floor selection* for orders in one pick box (task 3), including reservations on the selected floors. At this point, physical pick boxes are created to start their path through the warehouse on the conveyor belt. When the pick box arrives on the floor, the precise *product position selection* on the precomputed floor is determined (task 4), and corresponding SKUs' reservations are performed. The further procedure performs the *pick box batching* into picking carts (task 5), where several pick boxes are assigned to picking carts to be served together. Finally, pickers are assigned to picking carts, and a *picker routing* (task 6) is computed to walk through all positions where some reserved SKUs from the pick boxes in the picking cart are placed.

While such a decomposition helps understand the overall process, realizing some processes together, such as product position selection and picker routing, may be advantageous to allow for better solution quality. Another important thing to realize is related to our terminology since several hierarchical batching levels exist. First, we start with the batch selection to obtain a larger set of orders. Next, a grouping of orders from one batch is computed by order batching. Finally, the pick box batching groups pick boxes into picking carts.

Current Solution Approach

Our current work is oriented on higher-level processes aimed at order batching and floor selection for each pick box before sending it to the conveyor belt. To start these tasks, a batch selection is completed by taking a predefined number of orders sorted by the earliest limit time as a batch.

Further steps of the processing (tasks 4–6) are completed with the help of the simulation based on the average number of picks per picker, the number of pickers per floor, and the shift duration.

Order Batching

We have proposed three types of procedures for order batching into pick boxes. The first method aims to *maximize the number of single-floor (MSF)* pick boxes to minimize the number of floors where the pick box needs to be processed. This method must be combined with some further mentioned approaches since it can be used only for orders that can be fulfilled from a single floor. First, all single-floor options for each order are generated. Next, we queue orders for each floor sorted using their limit time. Note that some orders may be in several queues if they can be fulfilled from more single floors. We will create a pick box from the orders in the first random queue and remove selected orders from the remaining queues. The next pick boxes are created from other queues by round-robin to balance the workload.

Two other methods belong to the class of clustering algorithms (Aggarwal and Reddy 2014), clustering of orders into pick boxes. The second method is inspired by hierarchical agglomerative clustering with time-saving modification. The approach based on the Clarke and Wright algorithm (Gu, Goetschalckx, and McGinnis 2007) was developed for the vehicle routing problem (van Gils et al. 2018a). Here, the goal was to join orders into clusters/routes to minimize the picking distance in the warehouse. Our *Savings* method takes a batch of orders sorted by their limit time. We trivially compute all single-floor options for each order. If unavailable, a multi-floor option is generated for each order by solving the minimal hitting set (MHS) problem. To compute it, we apply the same procedure as we describe for floor selection for orders in one pick box (see the next section). The initial cluster set $Clusters$ contains one cluster for each order in the batch. Next, the distance for each of the two initial clusters is computed. The computation uses the minimal number of floors $fl(c)$ to fulfill orders in cluster c derived from the available floor options for each order using the procedure for MHS once more:

$$distance(c_1, c_2) = 2fl(c_1 \cup c_2) - fl(c_1) - fl(c_2) \ .$$

The equation tells how much the cluster $c_1 \cup c_2$ is worse than the cluster c_1 plus how much the cluster $c_1 \cup c_2$ is worse than the cluster c_2.

Iteratively, we take two clusters $a, b \in Clusters$ with the minimal distance such that the volume of $c = a \cup b$ fits into the pick box[1]. If the resulting cluster volume is sufficient[2], we remove a and b from $Clusters$, and c materializes into a new pick box. If insufficient, a and b are removed from $Clusters$, and c is added. Finally, the distance between c and remaining clusters from $Clusters$ is recomputed, introducing an incremental recomputation. Iterations are repeated until the number of orders in $Clusters$ reaches $MinBatch$ limit.

A more efficient but simpler version of the clustering is implemented by the *Seed* algorithm, which creates one cluster at a time only (van Gils et al. 2018a). Orders are again sorted by their limit time. We start with the first order as a new cluster. We go through the remaining orders fitting into pick boxes (using the $MaxVolume$ variable above) and select the order with the smallest distance from the cluster.

[1]We require that the total volume of all products in c must be at most $MaxVolume$ percents of the pick box volume.

[2]For now, the cluster c has sufficient volume if the total volume of all products in the cluster is at least $MinVolume$ percents of the pick box volume.

We add this order into the cluster. This process is repeated with remaining orders until we have a sufficient volume of the cluster (using the $MinVolume$ variable). The resulting cluster defines a new pick box. The overall process is repeated with the remaining orders from the batch until their number reaches $MinBatch$ limit.

Floor Selection

The floor selection aims to select the smallest set of floors for orders in the pick box to avoid unnecessary delays by visiting additional floors. To compute it efficiently, We apply the greedy algorithm for the *minimal hitting set (MHS)* problem (Arpino, Dmitriev, and Grometto 2023), the dual problem to the set covering (Johnson 1974). Initially, each order is associated with several single-floor options or one multi-floor option, which can be used to fulfill the order (see previous section). The goal is to find a set of floors covering a single- or multi-floor option for each order. In each step, we select the floor f with the largest sum of coverings for all orders.

Example: We have two orders with single-floor options $\{\{2\},\{3\},\{4\}\}$ and $\{\{2\}\}$ and one order with multi-floor option $\{\{1,2\}\}$. The floors 1,2, and 3 have the sum of coverings $0+0+0.5$, $1+1+0.5$, and $1+0+0$, respectively. So, floor 2 has the largest sum of coverings, 2.5.

We will remove f from all subsets of particular orders. If any subset is empty, the order is fulfilled by selected floors and is no longer considered. Once all orders are fulfilled, we have the final set of floors.

Preliminary Experimental Evaluation

The experiments were provided with the help of half-year data from the Notino company. For the current work, we have prepared six data sets with 117, 89, 75, 65, 56, and 46 thousand orders on average. Each data set contains 14 days of real processing taken randomly to have roughly decreasing average number of orders in each data set. If not mentioned otherwise, the results are measured in the middle 10 days to cut off the initial and final progress. The experiments were performed on Debian x86_64 Linux with AMD EPYC 7543 CPU (each run using 1 CPU core and 4 GB of RAM) in MetaCentrum distributed computing infrastructure in the Czech Republic. For each measurement, the average of 10 simulation runs is reported.

Each simulation starts with the random storage initialization (Masae, Glock, and Vichitkunakorn 2020) based on the products from the first three days, which serves us as an oracle. During the processing, restocking is completed every two hours with a small overstocking (20 % in our setup). One step of the simulation is processed every 1 minute, and order batching and floor selection is computed every 3 minutes. Each batch has 250 orders, and batching algorithms are repeated until less than 50 orders remain (see $MinBatch$ limit in the previous section). The volume of SKUs in pick boxes ranges between 60 % and 80 % of their total volume (see $MinVolume$ and $MaxVolume$). The further steps of processing on one floor are computed during the simulation with the help of 160 picks per picker on average for the two 8-hour long shift.

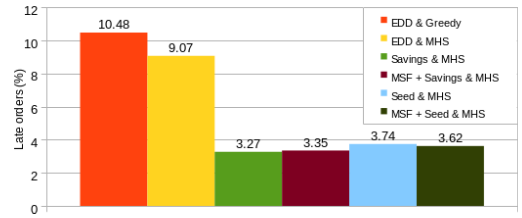

Figure 2: The number of late orders

For the order batching into pick boxes we apply the following methods:

- *Earliest Due Date* as a baseline, batching orders simply in order derived from their limit time;
- *Savings* clustering method having quadratic complexity based on the number of orders in the batch;
- *Seed* algorithm creating one cluster at a time;
- *MSF+Savings* applying maximization of single floors on each batch first and *Savings* on remaining orders;
- *MSF+Seed* applying maximization of single floors on each batch first and *Seed* on remaining orders.

For the floor selection for pick box fulfillment, the following methods are used:

- *Greedy* baseline approach selecting the most constraining floors for each order first (by ascending the number of floors containing different products and descending in the their quantity in pick box);
- *MHS* applying greedy procedure to solve the minimum hitting set problem.

To see the impact of different methods, we provide a comparison using the data set with 56 thousand orders per day, representing the average number of the orders (see Figure 2) . The baseline EDD & Greedy approach has 10.48 % late orders, which is only slightly improved by EDD & MHS (9.07 %). The performance of other approaches with some form of clustering was comparable. Seed & MHS and MSF+Seed & MHS achieved 3.74 % and 3.62 %, respectively. Savings & MHS and MSF+Savings & MHS were slightly better with 3.27 % and 3.35 %. The difference between the number of pickers among all methods was negligible (around ten pickers per day per floor) since the base computation was applied for all methods inferred from the number of orders for the current day and the orders remaining from the previous day.

The significant difference among the methods is visible in terms of CPU time (see Figure 3), which is provided for the complete simulation run including the first and last two days. Simulations took only 42 and 58 seconds for both EDD methods on average, with MHS slightly more demanding. With no surprise, Savings & MHS took the longest with 2,501 seconds. If single-floor orders do not go to clustering (MSF+Savings & MHS), the runtime decreases to 917 seconds. On the other hand, the MSF approach introduced a slowdown for the Seed clustering from 232 to 280 seconds since Seed does not need the quadratic complexity as Savings, and MSF resulted in the additional computation.

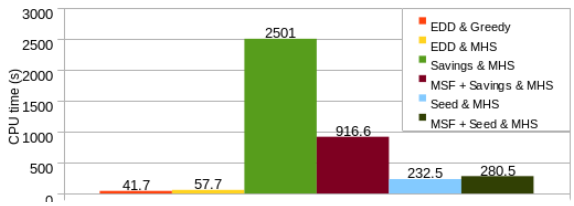

Figure 3: The CPU time of the whole simulation

Regarding the experiments over all data sets, it is essential to see a linear increase in the runtime going up to 4,700 seconds for the data set with the largest number of orders. This is undoubtedly because the batch size remains the same for all simulations. Since the most time Savings & MHS method also has the best performance in terms of the late orders, still being quite a simple heuristic approach, it is essential to realize that processing its simulation needs 0.35 seconds (4,700/14*16*60 for 14 days, 16 hours and 60 minutes) every one processing minute in average. This means there is a potential for a more complex method that is, at most, 100 times slower.

Conclusion and Future Work

We have introduced a complex problem for planning the warehouse processes with multiple floors. We have concentrated on handling the component for multiple floors, corresponding to the fact that it is a much less studied problem in the literature (Pardo et al. 2024). This ongoing work represents the first step in our study of this problem. We further concentrate on developing the methods for the single floor and storage manipulation and continue developing the heuristics for multiple floors. We are working on implementing the detailed simulator, which would encapsulate all the processes.

Acknowledgments

We want to thank the company Notino, which provided us with the production data from a half year of their processing. Our special thanks go to Martin Kavřík with his colleagues, who enabled us to understand warehouse processes in Notino and helped us obtain the Notino data.

Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

References

Aggarwal, C. C.; and Reddy, C. K. 2014. *Data clustering: Algorithms and applications*. CRC Press.

Arpino, G.; Dmitriev, D.; and Grometto, N. 2023. Greedy heuristics and linear relaxations for the random hitting set problem. *arXiv preprint arXiv:2305.05565*.

Casella, G.; Volpi, A.; Montanari, R.; Tebaldi, L.; and Bottani, E. 2023. Trends in order picking: a 2007-2022 review of the literature. *Production & Manufacturing Research*, 11: 1–74.

Chen, T.-L.; Cheng, C.-Y.; Chen, Y.-Y.; and Chan, L.-K. 2015. An efficient hybrid algorithm for integrated order batching, sequencing and routing problem. *International Journal of Production Economics*, 159: 158–167.

Gu, J.; Goetschalckx, M.; and McGinnis, L. F. 2007. Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, 177: 1–21.

Johnson, D. S. 1974. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3): 256–278.

Masae, M.; Glock, C. H.; and Vichitkunakorn, P. 2020. Optimal order picker routing in a conventional warehouse with two blocks and arbitrary starting and ending points of a tour. *International Journal of Production Research*, 58(17): 5337–5358.

Pardo, E. G.; Gil-Borrás, S.; Alonso-Ayuso, A.; and Duarte, A. 2024. Order batching problems: Taxonomy and literature review. *European Journal of Operational Research*, 313: 1–24.

Scholz, A.; Schubert, D.; and Wäscher, G. 2017. Order picking with multiple pickers and due dates - Simultaneous solution of order batching, batch assignment and sequencing, and picker routing problems. *European Journal of Operational Research*, 263(2): 461–478.

van Gils, T.; Ramaekers, K.; Braekers, K.; Depaire, B.; and Caris. 2018a. Increasing order picking efficiency by integrating storage, batching, zone picking, and routing policy decisions. *International Journal of Production Economics*, 197: 243–261.

van Gils, T.; Ramaekers, K.; Caris, A.; and de Koster, R. B. 2018b. Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review. *European Journal of Operational Research*, 267(1): 1–15.

Van Kampen, T. J.; Akkerman, R.; and Pieter van Donk, D. 2012. SKU classification: a literature review and conceptual framework. *International Journal of Operations & Production Management*, 32(7): 850–876.

Vanheusden, S.; Van Gils, T.; Ramaekers, K.; Cornelissens, T.; and Caris, A. 2023. Practical factors in order picking planning: state-of-the-art classification and review. *International Journal of Production Research*, 61(6): 2032–2056.

Weidinger, F. 2018. Picker routing in rectangular mixed shelves warehouses. *Computers & Operations Research*, 95: 139–150.

Yu, M.; and de Koster, R. B. 2009. The impact of order batching and picking area zoning on order picking system performance. *European Journal of Operational Research*, 198(2): 480–490.