Computing Optimized Stock (Re-)Placements in Last-In, First-Out Warehouses*

Ulrike Ritzinger, Matthias Prandtstetter, and Günther R. Raidl

Institute of Computer Graphics and Algorithms
Vienna University of Technology, Vienna, Austria
`u.ritzinger@rat.at, {prandtstetter|raidl}@ads.tuwien.ac.at`

Abstract. Within this work, we focus on the optimization of storage location assignments arising in warehouses with storage locations applying a last-in, first-out throughput policy. The sequence of goods to be stored is, however, not entirely known such that for each item the currently best storage location has to be identified almost immediately. Caused by this imperfect data and by stock removals concurrently performed it is necessary to apply relocation operations from time to time, which might range from a few operations to relocations lasting a working day depending on the workload of the warehousemen. For this purpose we propose an ad hoc stocking strategy as well as a storage relocation strategy based on variable neighborhood descent. Supported by experimental tests we compare variants of our approaches with each other and with formerly used stocking strategies showing that the number of conflicts could be significantly reduced by the proposed approach. Furthermore, an application of the relocation strategy can significantly improve warehouse states obtained due to imperfect stocking strategies, concurrently performed stock removals and insufficient information on production sequences.

1 Introduction

Nowadays, warehouse management is beside supply chain management one of the most important tasks in production environments. On the one hand it has to be assured that all components needed during the production process are available but on the other hand the costs induced by storage capacity, i.e., storage space, should be minimized. Furthermore, the access times to individual items located in the storage should be kept as low as possible in order to optimally serve customers. For collecting ordered items, it is of interest to minimize the access times by computing short routes through the warehouse [1, 2]. Yet, these tours are strongly dependent on the layout structure of the warehouse. Therefore, a reorganization of the storage exploiting properties of the goods to be stored and typical customer behavior [3] may be in some situations more convenient.

Anyhow, about 33 percent of money invested in logistics can be attributed to the costs arising in inventory management [4]. Therefore, a proper investigation

* This work was kindly supported by Dataphone GmbH, Vienna, Austria, and Hamburger-Spremberg, Spremberg, Germany.

of savings that might be achieved within this part of supply chains is necessary and in many cases profitable, see [4] for a literature review of the work on this topic over the last 30 years.

One of the main factors in the organization of warehouses is the throughput policy to be used within the storage [5]. The best known strategies are last-in, first-out (LIFO) and first-in, first-out (FIFO) policies. There are also other policies especially developed for different kinds of goods to be stored, e.g., first-produced, first-out (FPDO) or first-expire, first-out and first-deliver, first-out (FDFO). FPDO and FDFO mainly find application in lines of businesses coping with products having best-before dates assigned [5]. While these main inventory decisions have to be made when building warehouses, it often has to be decided on a daily or even shorter basis, which articles to be placed at which locations.

In this work, we focus on such a storage location problem arising in paper industry. To optimize the production efficiency (e.g., minimize cutting loss) it is common to temporarily store the just produced paper rolls in an intermediate store until the customers pick up their orders. Obviously it is desired that this process should be performed as fast as possible. Therefore, it is highly preferable that all paper rolls of currently served customers are directly accessible. If this is not the case the according rolls should be reinserted into the storage at the best possible location. Especially at the end of a working day there might be time left for reordering the complete warehouse such that all orders to be processed during the next day are optimally accessible.

In the next two sections a detailed description of the underlying problem and a more formal definition including an evaluation function for estimating the likelihood of additional paper roll relocations during stock removal are given. Based on this evaluation function a stocking strategy is presented in Section 4, and Section 5 includes the description of relocation strategies utilizing variable neighborhood descent and greedy methods. Both approaches are evaluated on different warehouse states in Section 6. Conclusions including future work are finally drawn in Section 7.

2 Production Process

In paper industry it is common to apply a three stage production process: At first paper roll blanks are produced which are cut into paper rolls of individual lengths in the second stage. At last the produced rolls are shipped.

In our case, the first two steps of the production process are optimized such that all ordered rolls of same paper type and grammage are consecutively produced and that the offcut of each blank is as small as possible. Due to this the production order of rolls is not sorted according to any attribute which is relevant for shipping. After cutting the paper rolls are transported on a conveyor belt into the warehouse.

The warehouse itself is organized as follows: The storage consists of parallel aisles and each aisle contains to its left and right side storage locations for storing rolls, also called strips (see Fig. 1a). Although, there is no structural separation

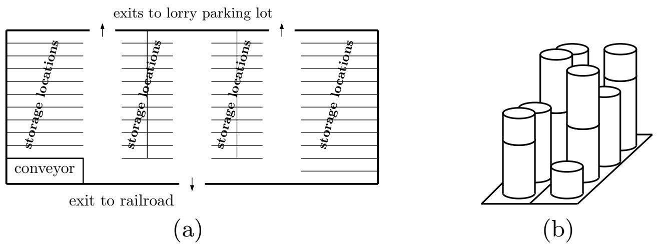

Fig. 1: (a) Schematic plan of the warehouse and (b) extract of the storage.

between the strips they are always accessed from the corresponding aisle. In each strip the first roll is placed at the end of it and all further rolls are put in according to the last-in, first-out throughput policy, see Fig. 1b for an extract of two strips with rolls stored at them. The strips are all physically identical except with respect to their capacities. But there are some strips dedicated to the storage of rolls of short length, i.e., rolls with less than 1250 mm. Such rolls can be loaded in a more space saving way and thus they should be stored together at theses special locations. There are three workers (one is responsible for placing the rolls from production and two are assigned for picking up outgoing rolls) who are equipped with forklift trucks for transporting the rolls, mobile terminals displaying various information (e.g., where the roll should be stored) and mobile bar code scanners to communicate each operation (removal and placing of rolls) to a centralized database. Therefore, the current state of the warehouse including the advance in the loading of paper rolls is known at any time.

While the assignment of rolls to storage locations is done in a greedy manner it often occurs that rolls have to be rearranged to provide direct access to those rolls for shipping. This leads to increased removal times. Therefore, it is important that the removal sequence of the rolls is considered at the time of storage. The most important criteria therefore are the shipping dates of the rolls. Besides this there are also other attributes: It is desired that the rolls are grouped by customers, i.e., all rolls for one customer should be at the same location. In addition, each customer has a preferred type of shipment (by lorry or freight car) and therefore the rolls should be placed in those strips which are near the according exit, see also Fig. 1a. For a complete list of constraints and different aims regarded in this work and their influence on an evaluation function used for our optimization approach, see Section 3.

Unfortunately, it is in our case not possible—neither for the worker nor for any computer aided decision system—to gather the exact production sequence. This is mainly caused by frequently breakdowns or failures of certain machines needed for special types of paper such that for a short term other paper types are produced or express orders of customers which have to be fulfilled almost immediately. Thus it has to be mentioned that the investigated storage location assignment problem is an *online problem* [6]. A further reason for not globally planning a fixed storage location for each roll is that customers frequently pick up their order lately or even too early.

Therefore, the depositer has to estimate the best available storage location for each roll to be stored. In literature there exists some work for the related *storage location assignment problem* [7–10], which has been shown to be $\mathcal{NP}$-hard [8]. Anyhow, according to the classification used in [7] the storage location assignment problem examined within this work uses class-based storage as stock location assignment strategy, i.e., each roll is assigned to a certain class according to its attributes and then (arbitrarily) stored at a location dedicated to that class of paper rolls. To overcome this arbitrariness in selecting the best available storage location for each paper roll, we propose a more fine grained evaluation function within this work that assigns to each (possible) warehouse state a positive value approximately indicating the likelihood of occurring conflicts, i.e., paper roll reallocations, during stock removals. The most promising paper roll assignments are suggested to the warehousemen via a mobile terminal.

Although this approach works quite well, it must not be disregarded that due to stock removals there might arise the situation that an explicit rearrangement of paper rolls significantly improves the warehouse situation, i.e., reduces the number of conflicts in future stock removals. Therefore, we additionally use a *variable neighborhood descent* [11] based approach for computing rearrangement operations that can be performed by warehousemen currently not busy. Again, there are some requirements which have to be regarded when implementing such a method. Mainly, it has to be assured that after each single rearrangement operation, the warehouse state is reasonable good such that in case the rearrangements have to be suspended the stock removal operations can still be efficiently performed. Due to the fact that rearrangement operations are only performed when no other jobs are to be completed it is not possible to count on these reallocations during storage location assignment. In addition, the number of times one roll is moved should be kept low.

3 Problem Definition

Within this section we provide a more formal definition of the introduced problem. In addition we present an evaluation function which will in the further context be used for approximately indicating the likelihood of conflicts, i.e., necessary paper roll reallocations, arising during stock removal operations.

We are given a warehouse W and a set of n_r paper rolls $R = \{1, \ldots, n_\mathrm{r}\}$, which includes all rolls in the system. The warehouse itself consists of n storage locations $i \in W = \{1, \ldots, n\}$, which are organized according to a last-in, first-out throughput policy. Therefore, we can define a tuple $S_i = (s_{i,1}, \ldots, s_{i,f_i})$ for each strip $i \in W$ indicating that roll $s_{i,l}$ has been assigned to i before $s_{i,l+1}$, with $1 \leq l < f_i$ and $f_i \in \mathbb{N}_0$ indicating the fill level of strip i, i.e., the number of rolls stored in storage location $i \in W$. While each strip $i \in W$ has a maximum capacity c_i, each paper roll $j \in R$ has a given weight w_j. Obviously, at each time $\sum_{j \in S_i} \mathrm{w}_j \leq \mathrm{c}_i$ holds. Further we are given a set of n_o orders O stated by costumers, where each order $K \in O$ is a set $K \subseteq R$ of paper rolls and define the set $\Omega(i) = \{K \in O \mid \exists\, l : s_{i,l} \in K\}$ as the set of all orders having at least one paper roll $l \in R$ stored in strip $i \in W$. In addition, we define set $D_K = \{i \in W \mid S_i \cap K \neq \emptyset\}$ as the set of storage locations containing at least one roll of an order $K \in O$. As already mentioned the exact shipping date is not known, but an expected shipping date d_K as well as the preferred shipping mode m_K are given for each order $K \in O$.

Each paper roll $j \in R$ has a certain positive length, and those rolls with a length shorter than 1250 mm are called *small goods*. Constants $\mathrm{w}_i^{\mathrm{small}} \in \{0, 1\}$ indicate which storage locations $i \in W$ are dedicated to storing small goods; for these locations $\mathrm{w}_i^{\mathrm{small}}$ is one and for all others zero. Similarly, constants $\mathrm{w}_i^{\mathrm{truck}} \in [0, 1]$ and $\mathrm{w}_i^{\mathrm{train}} \in [0, 1]$ define with which preference paper rolls shipped by truck or train should be stored at locations $i \in W$, respectively. Note that $\mathrm{w}_i^{\mathrm{truck}} + \mathrm{w}_i^{\mathrm{train}} = 1$ does not necessarily hold. Finally, we define a (current) warehouse state $\mathcal{W}$ as a snapshot of the current situation in the warehouse.

Evaluation Function

To present a method for either finding the best storage location(s) for a given paper roll to be stored or moving operations for improving the situation in the warehouse within this work, it is necessary to develop an evaluation function that approximately indicates the likelihood of conflicts, i.e., paper roll reallocations, during stock removals for a given warehouse state $\mathcal{W}$. The basic concept of this evaluation function $\mathrm{E}(\mathcal{W})$ is as follows: In case $\mathrm{E}(\mathcal{W}) = 0$ holds it is very likely that during stock removals no additional reallocations of paper rolls are necessary. With increasing value of $\mathrm{E}(\mathcal{W})$ this likelihood decreases, i.e., the likelihood of reallocations increases. In addition, not only the likelihood increases but also the expected number of occurring conflicts, i.e., more reallocations will become necessary during each stock removal step. Therefore the value of $\mathrm{E}(\mathcal{W})$ is not (strongly) bounded from above, since it is almost always possible to generate a worse warehouse state by adding an additional roll to the storage that generates additional conflicts. The evaluation process, however, is based on restrictions implied by observations stated by the warehouse manager of our paper production company.

The evaluation of a current warehouse state $\mathcal{W}$ is done in two steps: firstly all possible conflicts arising in single strips $i \in W$ are computed and secondly a rating regarding the complete warehouse is done. Both values are then appropriately weighted and the sum represents the objective value of the warehouse state. The most important reason for conflicts is the expected shipping date d_K of the orders $K \in O$. Therefore, we introduce function $\mathrm{date}(i) \geq 0$ counting the number of conflicts in strip $i \in W$ occurring with respect to the shipping dates associated with the rolls stored in i. A conflict occurs for two paper rolls $s_{i,l}$ and $s_{i,l'}$, with $s_{i,l}, s_{i,l'} \in S_i$, iff $l < l'$ and $\mathrm{d}_K < \mathrm{d}_{K'}$, with $s_{i,l} \in K$, $s_{i,l'} \in K'$, respectively, and $K, K' \in O$, i.e.,

$$\mathrm{date}(i) = \sum_{l=1}^{f_i - 1} \sum_{l'=l+1}^{f_i} \chi^{\mathrm{d}}(s_{i,l}, s_{i,l'}), \qquad \forall i \in W, \tag{1}$$

with $\chi^{\mathrm{d}}(j, j') = 1$ iff roll j is going to be shipped before j'; otherwise $\chi^{\mathrm{d}}(j, j') = 0$, with $j, j' \in R$.

Another important reason for conflicts is the mix-up of different orders within one strip. Therefore function order(i) counts the number of different orders stored at location $i \in W$; in addition the inhomogeneity of orders stored in the same strip is considered, yielding the following definition:

$$\mathrm{order}(i) = |\Omega(i)| + \sum_{l=1}^{f_i - 1} \chi^{\mathrm{o}}(s_{i,l}, s_{i,l+1}), \qquad \forall i \in W, \tag{2}$$

with $\chi^{\mathrm{o}}(j, j') = 1$ iff the orders of rolls $j, j' \in R$ are different; otherwise $\chi^{\mathrm{o}}(j, j') = 0$. Next, the number of strips used for storing all rolls of an order $K \in O$ is computed. Function distr$(\mathcal{W})$ sums up these distribution values of all orders:

$$\mathrm{distr}(\mathcal{W}) = \sum_{K \in O} D_K. \tag{3}$$

Since the orders of customers are known, it can be decided if there are still some paper rolls in production or not. Of course, it should be emphasized that each order is stored at few locations (the best case is if only one strip is needed for each order). Therefore, it is meaningful to reserve space in the storage for each uncompleted order. This is done by using function cap$(\mathcal{W})$ which computes the number of paper rolls not yet stocked and which cannot be assigned to the same storage location as the other paper rolls of the same order:

$$\mathrm{cap}(\mathcal{W}) = \sum_{K \in O} \max \left\{ \min_{i \in W} \left\{ \left| K \setminus \bigcup_{i' \in W} S_{i'} \right| - \left(\mathrm{c}_i - \sum_{j \in S_i} \mathrm{w}_j \right) \right\}, 0 \right\} \tag{4}$$

In addition, function compl$(\mathcal{W})$ counts the number of orders that are not yet completely stocked and have at least one stocked roll blocked by another one of another order. Thus, blocking of not yet completely stocked orders is also penalized.

Since the process of loading the rolls on lorries or freight cars should be finished as fast as possible (by minimizing the lengths of the paths the warehousemen have to move the rolls). Therefore, we store paper rolls preferably close to the exit presumably later used during stock removal. This is done according to function ship(i), with $i \in W$, under the assumption that v_j^{t} is equal to one iff roll $j \in R$ should be shipped with trucks; otherwise $v_j^{\mathrm{t}} = 0$ holds:

$$\mathrm{ship}(i) = \sum_{j \in S_i} \left(v_j^{\mathrm{t}} \cdot \left(1 - \mathrm{w}_i^{\mathrm{truck}}\right) + \left(1 - v_j^{\mathrm{t}}\right) \cdot \left(1 - \mathrm{w}_i^{\mathrm{train}}\right) \right), \qquad \forall i \in W. \tag{5}$$

Analogously, function small(i), with $i \in W$, increases when long paper rolls are stored in strips dedicated to small goods or when short paper rolls are stored in strips not dedicated to small goods. Under the assumption that $v_j^{\mathrm{s}} = 1$ for paper rolls $j \in R$ with a length shorter than 1250 mm (and $v_j^{\mathrm{s}} = 0$ in other cases), small(i) can be defined as:

$$\mathrm{small}(i) = \sum_{j \in S_i} \left| v_j^{\mathrm{s}} - \mathrm{w}_i^{\mathrm{small}} \right|, \qquad \forall i \in W. \tag{6}$$

Empty strips are the most valuable ones because virtually all paper rolls or even orders can be stored in them while increasing the objective function only minimally, if at all. Therefore, it should be well considered at which time empty strips will be started to be used. For this, we define a function empty(i) increasing the objective function by a small amount if strip $i \in W$ is not empty, i.e.,

$$\mathrm{empty}(i) = \begin{cases} 0 & \text{if strip } i \text{ is empty} \\ 1 & \text{otherwise} \end{cases} \tag{7}$$

Finally, it is possible to define the evaluation function $\mathrm{E}(\mathcal{W})$ which combines all previously defined functions. Each of the sub-functions is weighted using an appropriate coefficient in order to balance the influence of the different components among each other:

$$\begin{aligned} \mathrm{E}(\mathcal{W}) = \sum_{i \in W} \Big(& \gamma^{\mathrm{d}} \cdot \mathrm{date}(i) + \gamma^{\mathrm{o}} \cdot \mathrm{order}(i) + \\ & \gamma^{\mathrm{s}} \cdot \mathrm{ship}(i) + \gamma^{\mathrm{e}} \cdot \mathrm{empty}(i) + \gamma^{\sigma} \cdot \mathrm{small}(i) \Big) + \\ & \gamma^{\delta} \cdot \mathrm{distr}(\mathcal{W}) + \gamma^{\kappa} \cdot \mathrm{cap}(\mathcal{W}) + \gamma^{\mathrm{c}} \cdot \mathrm{compl}(\mathcal{W}) \end{aligned} \tag{8}$$

Though objective function (8) covers the most important aspects during stocking operations in our particular paper industry application, there are further special cases that can be considered. For a more detailed approach we refer to [12].

4 Stocking Strategy

Based on function (8) it is easy to develop a straight forward greedy stocking strategy. For this purpose, one simply needs to compute the changes in $\mathrm{E}(\mathcal{W})$ when adding a roll to the storage. After doing this for all strips that one resulting in the best warehouse state is chosen. For pseudocode of this procedure see Alg. 1.

Influence of the Weighting Coefficients

As already mentioned above each of the sub-functions of the objective function (8) is weighted by a factor for controlling its influence on the evaluation of a given warehouse state. Unfortunately, it is not trivial to find a parameter setup being valid for any production setting. Even more, there exists no generally good weighting factor adjustment. Although this might be disappointing for warehouse operators, this circumstance holds a crucial advantage: By tuning

Algorithm 1: Stocking Strategy

```
Input: $\mathcal{W}$: current warehouse state, $j \in R$: paper roll
Data: $bestStrip$: so far best strip for paper roll $j$,
      $bestEval$: value of best so far found warehouse state
Output: strip $i \in W$ roll $j$ should be assigned to

$bestStrip \leftarrow$ null ; $bestEval \leftarrow \infty$;
foreach $i \in W \setminus \{i'\}$ do
    $\mathcal{W}' \leftarrow \mathcal{W}$ after adding paper roll $j$ to strip $i$;
    if $\mathrm{E}(\mathcal{W}') < bestEval$ then
        $bestStrip \leftarrow i$;
        $bestEval \leftarrow \mathrm{E}(\mathcal{W}')$;
return $bestStrip$;
```

these parameters and adapting the relations it is possible to implement different stocking strategies. For example it might be promising to ensure for certain customers that paper rolls ordered by them are directly accessible all the time. In this case one will increase the weighting factor γ^{o} such that a mix up of orders becomes very unlikely. Nevertheless, this behavior might not be appropriate for all customers. In such a case the weighting factors may even be differently instantiated in dependence of the customers which finally leads to a more complex but also significantly more flexible objective function.

In this work, we use the following fixed weighting factors which was determined after consulting the warehouse manager of our case company: $\gamma^{\mathrm{d}} = 150$, $\gamma^{\mathrm{o}} = 5$, $\gamma^{\mathrm{s}} = 1$, $\gamma^{\mathrm{e}} = 20$, $\gamma^{\sigma} = 40$, $\gamma^{\delta} = 25$, $\gamma^{\kappa} = 20$, $\gamma^{\mathrm{c}} = 50$.

5 Relocation Strategy

Due to the online aspect of our problem, even the best stocking strategy finally results in suboptimal storage situations, which means that reallocations are necessary during shipment. Additionally, empirical data provided by our case company implies that the filling level of the storage usually is about 70–80%, i.e., the stocking opportunities are rather limited. When applying the above presented stocking strategy over a longer time, it is only able to prevent major immediate conflicts to a certain degree any more. For generally improving the warehouse state and to bypass idle times of warehousemen it is possible to perform relocations of paper rolls. Of course, the aim of this is to reduce the number of conflicts occurring during stock removals and improve the warehouse situation.

There are different types of possible relocations: The first class of reallocations is automatically to be performed during stock removal operations when one or more paper rolls are blocking rolls to be shipped. The second and more laborious type is performed during idle times of warehousemen. Dependent on the available time various movements can be done. To be flexible in this point the system has to accept inputs from the workers indicating the number of rolls to

```
Algorithm 2: GreedyRelocationProcedure($\mathcal{W}$, $n_\mathrm{m}$)
  Input: $\mathcal{W}$: current warehouse state, $n_\mathrm{m}$: number of available relocation moves
  Data: $\mathcal{W}'$, $\mathcal{W}''$: intermediate warehouse states
  Output: L: list of moves to be performed
  repeat
      $j \leftarrow \arg\min_{j' \in \bigcup_{i' \in W} \left\{ s_{i',f_{i'}} \in S_{i'} \right\}} \{\mathrm{E}(\mathcal{W}') \mid \mathcal{W}' = \mathcal{W}$ after removing roll $j'\}$;
      $\mathcal{W}' \leftarrow \mathcal{W}$ after removing $j$;
      $i \leftarrow \arg\min_{i' \in W} \{\mathrm{E}(\mathcal{W}'') \mid \mathcal{W}'' = \mathcal{W}'$ after storing roll $j$ at location $i'\}$;
      $\mathcal{W}'' \leftarrow \mathcal{W}'$ after moving $j$ to strip $i$;
      if $\mathrm{E}(\mathcal{W}'') < \mathrm{E}(\mathcal{W})$ then
          $\mathcal{W} \leftarrow \mathcal{W}''$;
          add appropriate movement instructions to L;
          $cnt \leftarrow cnt + 1$;
  until $cnt \geq n_\mathrm{m}$ or no further improvement could be achieved ;
  return L;
```

be reallocated resulting in a list of movements improving the current warehouse state.

One obvious approach for achieving this is a greedy method presented in the next section selecting always the next best paper roll for relocations. In addition, we present a variable neighborhood descent based approach generating movement lists to be processed by the warehouse workers.

5.1 Greedy Reallocation

The *greedy reallocation procedure* (GRP) removes a roll $j \in R$ which is directly accessible and causes conflicts during removal operations with highest probability, i.e., under the assumption that $\mathcal{W}$ denotes the current warehouse state

$$j = \arg\min_{j' \in \bigcup_{i \in W} \left\{ s_{i,f_i} \in S_i \right\}} \{\mathrm{E}(\mathcal{W}') \mid \mathcal{W}' = \mathcal{W} \text{ after removing roll } j'\} . \quad (9)$$

Afterwards this roll is reinserted into the storage at the best strip $i \in W$, i.e.,

$$i = \arg\min_{i' \in W} \{\mathrm{E}(\mathcal{W}'') \mid \mathcal{W}'' = \mathcal{W}' \text{ after storing roll } j \text{ at location } i'\} , \quad (10)$$

where $\mathcal{W}'$ denotes the warehouse state after removing paper roll j from its current strip. This procedure is repeated until either the number of available moves n_m is reached or there is no further improvement achievable, i.e., roll j is best stored at its original storage location. An outline of the pseudocode is given in Alg. 2. For experimental results obtained using GRP we refer to Section 6.

5.2 Variable Neighborhood Descent Based Approach

Obviously, the main disadvantage of GRP lies within the fact that moves are selected on a purely greedy basis disregarding the improvement potential of moves to be investigated in further steps. Thus, it is not possible or at least very unlikely to resolve conflicts arising in connection with paper rolls not directly accessible. We propose an approach based on *variable neighborhood descent* (VND) [11].

VND itself basically exploits the observation that any global optimum is also locally optimal with respect to any given neighborhood. Therefore, a successful application of VND mainly relies on appropriately defined neighborhood structures systematically examined. For this purpose, the first neighborhood structure is searched through until no further improvement can be achieved. Then, the next neighborhood structure is examined, but as soon as an improvement could be achieved the search is continued using the first neighborhood structure again. This is repeated until a solution is found that is locally optimal with respect to all previously defined neighborhood structures.

This implies that two criteria must be regarded when following a VND based approach: on the one hand the proper definition of neighborhood structures to be used and on the other hand an appropriate order for examining these neighborhood structures. Both aspects will be discussed in more detail in the following.

Neighborhood Structures The main idea of the neighborhood structures used in this work is to resolve conflicts with respect to the sub-functions of (8) step by step. The neighborhood structures are defined as follows:

N_1: Within this first neighborhood structure the above presented greedy reallocation procedure is applied to a given warehouse state $\mathcal{W}$, i.e., in each iteration that roll is moved which increases $\mathrm{E}(\mathcal{W})$ the most.

N_2: A new warehouse state $\mathcal{W}' \in N_2(\mathcal{W})$ of a given warehouse state $\mathcal{W}$ is obtained from $\mathcal{W}$ by first removing all rolls from a strip $i \in W$ such that no conflicts with respect to the shipping dates occur in strip i, see Eq. (1). Afterwards the removed rolls are assigned immediately to other strips $i' \in W \setminus \{i\}$.

N_3: When examining a neighborhood based on this neighborhood structure only the last assigned rolls $j \in S_i$ of a strip $i \in W$ are removed such that all of these rolls are contained in the same order $K \in O$ and the first not removed roll is not part of that order.

N_4: With this neighborhood structure it is tried to merge one order $K \in O$ distributed over more than one strip into one strip holding at least one paper roll $j \in K$. Anyhow, only rolls directly accessible are considered for moving.

N_5: To eliminate conflicts according to Eq. (2), we introduce this neighborhood structure which removes (and immediately reassigns) all rolls of a strip $i \in W$ except for the first to i assigned rolls, all contained in the same order, i.e., after removing the rolls $S_i \subseteq K \in O$ holds. Further, it is tried to completely empty a strip containing two or more mixed up orders.

N_6: This neighborhood structure is defined for resolving conflicts caused by rolls placed deep inside of any strip, i.e., rolls which were assigned to this storage location relatively early. Therefore, all paper rolls of a strip $i \in W$ are removed and then assigned to other strips $i' \in W \setminus \{i\}$ in the same order they were removed. Obviously, each solution contained within a neighborhood based on N_6 contains at least one strip which is totally empty.

While the size of N_1 is in $O(1)$, the sizes of all other neighborhoods is in $O(|W|)$. The time needed for examining N_1 is bounded from above by $O(|W|^2)$. Neighborhood structure N_3 can be searched in $O(|W| \cdot \max_{K \in O} \{|K|\})$. The examination time for all other neighborhood structures is in $O(|W| \cdot \max_{i \in W} \{|S_i|\})$. In order to efficiently examine each neighborhood, an incremental update of the evaluation function is implemented and the following two step functions are used:

Worst Neighbor: This step function selects among all solutions in a certain neighborhood $N(\mathcal{W})$ that one which rearranges those rolls causing the most conflicts with respect to $\mathrm{E}(\mathcal{W})$. Although very similar, preliminary tests revealed that a *best improvement* strategy is not as promising as this worst neighbor step function. Anyhow, in case of ties the first found is selected.

Random Neighbor: When using this step function for examining a neighborhood $N(\mathcal{W})$ one of the candidate solutions contained in $N(\mathcal{W})$ is chosen randomly. The probability for choosing one solution is proportional to the contribution of the moved rolls to the objective function (8), i.e., a roulette wheel selection is applied. Therefore, it is very likely to relocate rolls causing conflicts with high probability. At the same time, not so promising candidate solutions might also contribute to a finally computed list of relocation moves.

Neighborhood Order Beside the definition of neighborhood structures the sequence to be followed when examining them is of crucial importance for VND based approaches. Although there exist rules of thumb for ordering the neighborhoods [11], empirical tests in [13, 14, 2] revealed that dynamically chosen neighborhood orderings significantly improve the finally obtained solutions for some applications. Therefore, we investigated four neighborhood ordering strategies:

Ordered: This is the classical neighborhood ordering strategy as described in [11], i.e., the neighborhoods are arranged according to increasing size and/or examination times. Although the asymptotic examination times are similar to each other in the worst case, the actual examination times experienced in practice are on average increasing for N_1 to N_6. Therefore, when using this neighborhood ordering, N_i is examined before N_{i+1} for $i = 1, \dots, 5$. All neighborhoods are examined according to a worst neighbor strategy.

Reversely Ordered: The contribution of neighborhoods with small indices is limited to resolving conflicts occurring for paper rolls quite recently assigned to storage locations. Since the available time for reallocations is rather short, it is very likely that conflicts for paper rolls assigned early will not be resolved during relocation phases. In addition, by first applying more time expensive rearrangements, it is possible to eliminate these conflicts. Therefore, this neighborhood ordering first examines neighborhood structure N_6 and continues with N_5, N_4, N_3, N_2 and N_1, respectively. Again, all neighborhoods are searched using the worst neighbor step function.

Randomized: Based on the observation that the current warehouse state is permanently changing and therefore the type of arising conflicts is always in flux, a randomly chosen and constantly altering neighborhood ordering might be promising. Additionally, a variation in the utilized step function is performed such that there are nine different combinations of neighborhood structures and step functions. The application of the random neighbor strategy to neighborhood structures N_2, N_3 and N_6 will be denoted by N_7, N_8 and N_9, respectively, in the following. The next neighborhood N_i to be examined, with $i = 1, \ldots, 9$, is selected on a purely random basis each time a neighborhood examination is finished. In addition, a neighborhood structure is removed, i.e., no longer examined, if it did not yield a new improved warehouse state within its five last consecutive examinations.

Dynamically Randomized: Analogously to the randomized ordering strategy the next neighborhood N_i to be examined, with $i = 1, \ldots, 9$, is chosen randomly when applying this ordering strategy. Again N_7 to N_9 denote the application of the random neighbor strategy to N_2, N_3 and N_6, respectively. The probabilities for selecting the neighborhoods, however, are adjusted each time a selection is performed. Detailed values for the applied probabilities will be given below. Analogously to the purely randomized ordering, neighborhood structures are removed as soon as five consecutive examinations did not provide improved warehouse states.

VND Framework Given the current warehouse state $\mathcal{W}$, a number of paper rolls to be relocated, and a preferred neighborhood ordering and examination strategy *strat*, Alg. 3 can be used for computing a list of paper roll relocations to be performed by warehousemen. After initializing all temporary variables, the main loop of the procedure is entered and is executed until either no more neighborhoods are left to be examined according to *strat* or the number of yet available paper roll movements is equal to or less than zero. In case of either the randomized or the dynamically randomized neighborhood ordering is chosen, this algorithm is repeatedly executed for 50 rounds. Although any arbitrary value could be chosen for the maximum number of rounds, 50 seems to be promising based on the observation that computation times on a standard PC for our paper production company are then acceptable, i.e., at most about three minutes. The list of rearrangements resulting in the best new warehouse state is then returned by the algorithm. The following four different variants of VND were implemented and compared with each other (for test results see Section 6):

Ordered VND (OVND): For this variant of VND the parameter *strat* is set to ordered (neighborhood order is fixed and worst neighbor strategy is used). Since the algorithm is deterministic only one round is performed.

Reversely Ordered VND (ROVND): While the number of still available moves n_{m} is greater than or equal to 80, this algorithm uses a fixed reversely ordered neighborhood ordering. As soon as n_{m} falls below 80, the order is reversed, i.e., the same order as for OVND is used. Analogously to OVND only one round needs to be performed in Alg. 3.

Randomized VND (RVND): For this setting the randomized neighborhood ordering is selected. Therefore, N_1 to N_6 are using a worst neighborhood

Algorithm 3: RelocationVariableNeighborhoodDescent($\mathcal{W}$, n_m, *strat*)

```
Input: $\mathcal{W}$ ... current warehouse state,
       $n_\mathrm{m}$ ... number of available paper roll movements,
       $strat$ ... neighborhood ordering strategy and step function to be used
Data: $l$ ... index of currently examined neighborhood structure,
      $n', n''$ ... remaining number of available paper roll movements,
      $\mathcal{W}', \mathcal{W}''$ ... intermediate warehouse states,
      $bestVal$ ... value of the best so far obtained warehouse state,
      $bestL$ ... list of rearrangements for reaching the best so far obtained
      warehouse state
Output: L ... list of moves to be performed

$bestL \leftarrow ()$;
$bestVal \leftarrow \infty$;
repeat
    L $\leftarrow ()$;
    $\mathcal{W}' \leftarrow \mathcal{W}$;
    $n' \leftarrow n_\mathrm{m}$;
    repeat
        $l \leftarrow$ index of neighborhood to be examined next according to $strat$;
        $\mathcal{W}'' \leftarrow$ examine neighborhood $N_l(\mathcal{W}')$ according to $strat$;
        if $\mathrm{E}(\mathcal{W}'') < \mathrm{E}(\mathcal{W}')$ then
            $n'' \leftarrow n' -$number of roll moves needed for obtaining $\mathcal{W}''$ from $\mathcal{W}'$;
            if $n'' \geq 0$ then
                add roll relocations for obtaining $\mathcal{W}''$ from $\mathcal{W}'$ to L;
                $\mathcal{W}' \leftarrow \mathcal{W}''$;
                $n' \leftarrow n''$;
    until no more neighborhoods left to be examined or $n' \leq 0$ ;
    if $\mathrm{E}(\mathcal{W}') < bestVal$ then
        $bestVal \leftarrow \mathrm{E}(\mathcal{W}')$;
        $bestL \leftarrow$ L;
until until 1 or 50 repetitions are reached (dependent on strat) ;
return L;
```

strategy and N_7 to N_9 are searched by the random neighbor step function. Hence, 50 rounds are performed in the algorithm and the list of relocation moves resulting in the best warehouse state is returned.

Dynamically Randomized VND (DRVND): This variant applies dynamically randomized neighborhood ordering. The concrete probability values are shown in Table 1. Each time, the number of still available moves n_m falls below a value indicated in the column labeled n_m, the probabilities for selecting neighborhoods N_1 to N_9 are adjusted to the values shown in the corresponding columns. These values were identified during preliminary tests and were then refined by the warehouse manager of our case company.

Table 1: Selection probabilities of the neighborhoods in DRVND in dependence on the number of still available moves n_m.

n_m	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9
<20	0.50	0.20	0.10	0.05	0.03	0.02	0.04	0.04	0.02
20–100	0.35	0.15	0.10	0.10	0.02	0.15	0.05	0.05	0.03
>100	0.05	0.05	0.05	0.20	0.15	0.30	0.05	0.05	0.10

6 Experimental Results

In our paper production company the stocking strategy presented in Section 4 is already applied in practice. Comparing warehouse states previously obtained by the old stocking strategy, which was mainly based on the experience of the warehousemen as well as the warehouse manager with warehouse states obtained after using the here proposed stocking strategy, it can be clearly seen that the situation in the warehouse significantly improved and therefore the time needed for shipping is reduced by a vast amount. Unfortunately, it is not possible to directly compare the old and the new stocking strategy with each other during real time operations. Therefore, we decided to simulate the stocking of typically produced paper rolls using the old and the new strategy.

Using this simulation based data it is possible to compare the efficiency and contributions of the relocation strategies proposed within this paper. In fact, the main parameter of a typical warehouse state is the number of rolls stored within the warehouse. In our case at most 4400 paper rolls can be stored in the warehouse. Expert knowledge indicates that a filling level of 80% constitutes the critical level for which any further stocked paper rolls will almost always cause conflicts—even in case of optimal placement. Therefore we tested our relocation approaches on warehouse states with 2500, 3000 and 3500 paper rolls stocked. All computations were performed on a single core of a Dual Opteron processor with 2.4GHz and 4GB of RAM. As underlying database storing all production and order relevant data an Oracle 10i database was used. Although, the number of rolls to be relocated, can be chosen arbitrarily, we tested our algorithm for 10, 20, 50, 100, 300, 500 and 700 relocation moves, which corresponds to approximately 10, 25 and 60 minutes as well as 2, 6, 8 and 12 hours of working time. Although it is relatively rare that one worker might relocate all day long this might occur on weekends when only few new paper rolls are produced and the driving of trucks on highways is prohibited, which is law in some European countries.

For testing purposes we have chosen six exemplary production data sets called w_1 to w_6 which were provided by our paper production company. The limiting factor for our stocking strategy as well as the VND approach are the number of paper rolls to be stored in the warehouse as preliminary tests revealed. While warehouses w_1 and w_2 contain 2500 paper rolls to be stored, warehouses w_3 and w_4 consist of 3000 paper rolls. Finally, w_5 and w_6 contain 3500 rolls.

For evaluating the performance of the stocking strategy we compared three different stocking approaches. The first one corresponds to a simulation of the

Table 2: Absolute values of E($\mathcal{W}$) for six different test instances. The values represent the objective values for the warehouse states obtained by a simulated human stocking strategy and our stocking strategy proposed in Section 4. In the last column a lower bound on the objective value for the warehouse states is given.

	simulated stocking	stocking strategy		sorted stocking
		avg.	std	
w_1	102635.0	12819.9	339.1	11807.0
w_2	130856.0	14683.0	413.0	13778.0
w_3	186835.0	23047.1	1101.4	18244.0
w_4	135203.0	17290.2	1098.9	14012.0
w_5	300881.0	49481.0	3018.7	35850.0
w_6	181877.0	64687.3	2525.2	33080.0

stocking strategy used in our paper production company during the last years. This strategy is mainly based on the experience of the warehouse operator and does not provide any type of forecast—neither with respect to the shipping dates nor regarding paper rolls to be produced in future. The corresponding objective values with respect to Eq. (8) are given in the first data column of Table 2. The second column of this table lists the mean results over 20 runs using slightly different production sequences of paper rolls obtained by our stocking strategy including standard deviations. The final column lists values obtained by first sorting all ordered rolls according to their shipping date and customer order and then stocking them using our stocking strategy. It can be clearly seen, that our stocking strategy outperforms the formerly used strategy. It has to be emphasized that an optimal warehouse state will almost never be reached in this real world application as long as a last-in, first-out throughput policy is applied and the production process is optimized disregarding the storage structure.

Anyhow, it is still necessary to perform relocations from time to time since the shipping dates are often not met by the customers. These conflicts cannot be foreseen even by the best stocking strategy. While Table 3 represents values obtained for applying the relocation strategies on warehouse states generated by the formerly used stocking strategy, the values presented in Table 4 correspond to results obtained by reassigning paper rolls in warehouses obtained by our stocking strategy. A value of 98% indicates that an improvement of two percent could be achieved, i.e., the estimated probability of conflicts during removal operations with respect to Eq. (8) could be reduced by 2%.

The following trend can be recognized: the more available time is dedicated to relocation operations the better the obtained warehouse states become. In addition the performance of GRP seems to be poorer for a larger number of available moves than those of the different VND variants. Although the former tendency is obvious, GRP could not improve further with more than 100 moves for warehouse states considered in Table 3. It is most interesting that this behavior seems to be independent of the number of rolls stored in the warehouse. To confirm this observation, we performed additional tests investigating especially

Table 3: For different number of relocation moves the relative values of the finally obtained warehouse states based on those obtained via the formerly used stocking strategy are presented. Mean values are averages over 20 runs (with standard deviations in parentheses). The last column presents p-values of Wilcoxon rank sum tests for the hypothesis that the mean values of DRVND are better than those of RVND.

		GRP	OVND	ROVND	RVND			DRVND			p-val
					best	mean	std	best	mean	std	
$n_{\mathrm{m}} = 10$	w_1	97.2%	97.2%	97.2%	97.3%	97.7%	(0.3%)	97.1%	97.2%	(0.0%)	<0.01
	w_2	96.2%	96.2%	96.2%	96.2%	96.6%	(0.3%)	96.2%	96.2%	(0.0%)	<0.01
	w_3	98.9%	98.9%	98.9%	98.9%	99.0%	(0.1%)	98.6%	98.7%	(0.1%)	<0.01
	w_4	98.8%	98.8%	98.8%	98.6%	98.9%	(0.1%)	98.6%	98.8%	(0.0%)	<0.01
	w_5	97.7%	97.7%	97.7%	99.2%	99.4%	(0.2%)	97.7%	97.7%	(0.0%)	<0.01
	w_6	98.8%	98.8%	98.9%	98.8%	99.0%	(0.2%)	98.7%	98.8%	(0.0%)	<0.01
$n_{\mathrm{m}} = 20$	w_1	95.0%	95.0%	95.0%	95.8%	96.3%	(0.3%)	94.7%	95.0%	(0.2%)	<0.01
	w_2	92.9%	92.9%	92.9%	94.3%	94.7%	(0.2%)	92.9%	92.9%	(0.1%)	<0.01
	w_3	97.9%	97.9%	97.9%	97.5%	98.2%	(0.2%)	97.6%	97.8%	(0.1%)	<0.01
	w_4	98.4%	98.4%	98.4%	97.8%	98.1%	(0.2%)	97.6%	97.8%	(0.1%)	<0.01
	w_5	96.7%	96.7%	96.7%	97.5%	98.4%	(0.3%)	96.3%	96.7%	(0.2%)	<0.01
	w_6	98.0%	98.0%	98.1%	97.5%	98.2%	(0.3%)	97.0%	97.8%	(0.3%)	<0.01
$n_{\mathrm{m}} = 50$	w_1	89.6%	89.6%	89.6%	92.0%	92.3%	(0.1%)	89.4%	90.0%	(0.5%)	<0.01
	w_2	87.6%	87.6%	87.6%	89.4%	89.9%	(0.3%)	86.5%	87.6%	(0.7%)	<0.01
	w_3	94.7%	94.7%	94.7%	95.1%	96.1%	(0.4%)	94.2%	94.7%	(0.3%)	<0.01
	w_4	97.1%	97.1%	97.1%	95.0%	96.5%	(0.6%)	94.4%	95.8%	(0.8%)	<0.01
	w_5	93.5%	93.5%	93.5%	94.4%	96.0%	(0.4%)	92.7%	93.5%	(0.7%)	<0.01
	w_6	97.6%	96.2%	95.8%	95.6%	95.8%	(0.2%)	94.9%	95.3%	(0.3%)	<0.01
$n_{\mathrm{m}} = 100$	w_1	82.7%	82.7%	86.5%	86.6%	88.1%	(0.7%)	82.4%	83.3%	(0.8%)	<0.01
	w_2	84.9%	82.4%	83.1%	82.8%	84.4%	(0.8%)	81.1%	82.5%	(0.9%)	<0.01
	w_3	92.7%	92.3%	93.7%	91.8%	92.2%	(0.3%)	91.1%	92.0%	(0.3%)	0.03
	w_4	97.0%	95.8%	91.9%	89.8%	90.7%	(0.4%)	89.8%	90.2%	(0.3%)	<0.01
	w_5	90.4%	90.4%	91.7%	90.2%	90.7%	(0.2%)	89.5%	90.3%	(0.3%)	<0.01
	w_6	97.6%	93.4%	94.5%	92.7%	93.6%	(0.4%)	92.0%	92.5%	(0.5%)	<0.01
$n_{\mathrm{m}} = 300$	w_1	77.3%	67.7%	69.1%	71.0%	73.4%	(1.4%)	64.0%	68.6%	(3.1%)	<0.01
	w_2	84.9%	68.4%	66.4%	67.1%	68.9%	(1.0%)	65.0%	66.6%	(1.0%)	<0.01
	w_3	92.7%	85.6%	83.8%	83.3%	84.2%	(0.9%)	80.2%	82.5%	(1.3%)	<0.01
	w_4	97.0%	95.8%	84.5%	76.9%	79.9%	(1.5%)	73.2%	78.3%	(3.0%)	0.01
	w_5	90.4%	85.4%	81.0%	77.7%	78.9%	(0.7%)	77.7%	78.8%	(0.8%)	0.31
	w_6	97.6%	92.0%	95.5%	83.8%	84.9%	(0.7%)	83.4%	84.7%	(1.0%)	0.21
$n_{\mathrm{m}} = 500$	w_1	77.3%	63.5%	61.2%	63.0%	65.7%	(1.7%)	57.3%	59.8%	(2.1%)	<0.01
	w_2	84.9%	52.7%	51.5%	53.5%	55.3%	(1.9%)	50.4%	52.0%	(1.4%)	<0.01
	w_3	92.7%	79.1%	76.7%	74.8%	76.3%	(1.0%)	71.6%	73.6%	(1.3%)	<0.01
	w_4	97.0%	96.5%	65.7%	66.4%	67.8%	(1.3%)	64.0%	66.1%	(1.6%)	<0.01
	w_5	90.4%	77.7%	72.8%	68.9%	70.7%	(1.2%)	68.3%	70.4%	(1.2%)	0.10
	w_6	97.6%	93.4%	96.0%	76.1%	79.0%	(2.1%)	74.7%	77.2%	(1.6%)	<0.01
$n_{\mathrm{m}} = 700$	w_1	77.3%	59.1%	50.6%	55.3%	57.7%	(1.9%)	50.0%	51.2%	(1.3%)	<0.01
	w_2	84.9%	42.0%	43.0%	41.4%	44.0%	(2.2%)	40.4%	42.0%	(2.3%)	<0.01
	w_3	92.7%	74.8%	66.8%	64.4%	66.4%	(1.5%)	60.3%	63.2%	(2.1%)	<0.01
	w_4	97.0%	95.8%	60.0%	57.6%	58.9%	(0.9%)	53.5%	56.5%	(2.0%)	<0.01
	w_5	90.4%	76.2%	72.9%	62.5%	64.8%	(1.5%)	62.5%	64.1%	(1.4%)	0.01
	w_6	97.6%	92.8%	96.0%	68.8%	72.7%	(3.0%)	68.5%	71.2%	(1.9%)	<0.01

Table 4: For different number of relocation moves the relative values of the finally obtained warehouse states obtained via the proposed stocking strategy are presented. Mean values are averages over 40 runs (with standard deviations in parentheses). The last column presents p-values of Wilcoxon rank sum tests for the hypothesis that the mean values of DRVND are better than those of RVND.

		GRP	OVND	ROVND	RVND			DRVND			p-val
					best	mean	std	best	mean	std	
$n_\mathrm{m} = 10$	w_1	100.0%	99.5%	99.4%	97.9%	98.7%	(0.2%)	97.8%	98.3%	(0.1%)	<0.01
	w_2	100.0%	99.8%	99.8%	98.2%	99.2%	(0.1%)	97.7%	98.8%	(0.1%)	<0.01
	w_3	100.0%	99.4%	99.3%	97.8%	98.8%	(0.1%)	97.8%	98.6%	(0.1%)	0.11
	w_4	99.9%	99.8%	99.8%	98.0%	98.7%	(0.1%)	97.9%	98.5%	(0.1%)	<0.01
	w_5	99.8%	99.6%	99.6%	98.1%	99.2%	(0.1%)	98.2%	99.2%	(0.1%)	0.48
	w_6	99.9%	99.5%	99.5%	98.3%	98.8%	(0.1%)	97.8%	99.0%	(0.2%)	0.93
$n_\mathrm{m} = 20$	w_1	100.0%	99.4%	99.3%	97.3%	98.2%	(0.1%)	97.2%	97.8%	(0.1%)	<0.01
	w_2	100.0%	99.7%	99.6%	98.0%	98.8%	(0.1%)	97.7%	98.4%	(0.1%)	<0.01
	w_3	100.0%	99.1%	99.0%	96.3%	98.4%	(0.1%)	97.0%	98.1%	(0.1%)	0.04
	w_4	99.9%	99.8%	99.8%	95.5%	98.1%	(0.3%)	61.9%	96.3%	(1.2%)	<0.01
	w_5	99.8%	99.4%	99.3%	96.6%	97.8%	(0.3%)	96.4%	97.9%	(0.2%)	0.67
	w_6	99.9%	99.5%	99.5%	96.5%	97.9%	(0.1%)	96.7%	98.2%	(0.2%)	1.00
$n_\mathrm{m} = 50$	w_1	100.0%	98.9%	98.7%	95.2%	96.9%	(0.1%)	95.0%	96.2%	(0.1%)	<0.01
	w_2	100.0%	98.5%	98.5%	96.0%	97.0%	(0.1%)	95.6%	96.7%	(0.1%)	<0.01
	w_3	100.0%	99.1%	99.1%	95.3%	97.9%	(0.2%)	93.9%	97.1%	(0.3%)	<0.01
	w_4	99.9%	99.8%	99.8%	92.7%	96.8%	(0.2%)	68.8%	93.3%	(1.2%)	<0.01
	w_5	99.8%	99.3%	99.3%	92.5%	94.9%	(0.3%)	93.3%	95.3%	(0.3%)	0.99
	w_6	99.9%	99.1%	99.2%	93.5%	96.0%	(0.3%)	93.5%	96.2%	(0.4%)	0.85
$n_\mathrm{m} = 100$	w_1	100.0%	98.3%	98.1%	92.7%	96.0%	(0.3%)	68.0%	92.3%	(1.9%)	<0.01
	w_2	100.0%	99.7%	99.7%	93.4%	96.6%	(0.3%)	93.2%	95.6%	(0.2%)	<0.01
	w_3	100.0%	99.1%	99.1%	92.3%	97.5%	(0.4%)	92.4%	96.1%	(0.5%)	<0.01
	w_4	99.9%	99.8%	99.9%	92.5%	96.9%	(0.4%)	69.2%	93.5%	(1.3%)	<0.01
	w_5	99.8%	99.3%	98.3%	87.6%	92.0%	(0.7%)	86.7%	91.9%	(0.7%)	0.46
	w_6	99.9%	99.0%	99.1%	89.1%	94.4%	(1.3%)	90.1%	93.7%	(0.5%)	0.25
$n_\mathrm{m} = 300$	w_1	100.0%	99.2%	99.2%	93.7%	96.2%	(0.3%)	84.8%	93.1%	(0.7%)	<0.01
	w_2	100.0%	99.7%	99.7%	93.6%	97.0%	(0.3%)	91.8%	95.3%	(0.5%)	<0.01
	w_3	100.0%	99.1%	99.1%	92.8%	97.3%	(0.4%)	90.9%	96.0%	(0.5%)	<0.01
	w_4	99.9%	99.8%	99.9%	91.3%	96.9%	(0.4%)	86.8%	93.8%	(0.7%)	<0.01
	w_5	99.8%	99.2%	97.8%	82.5%	91.1%	(0.9%)	81.3%	88.7%	(0.6%)	<0.01
	w_6	99.9%	99.0%	98.4%	81.7%	91.2%	(1.9%)	73.8%	86.4%	(2.3%)	<0.01
$n_\mathrm{m} = 500$	w_1	100.0%	99.2%	99.2%	93.3%	96.2%	(0.3%)	85.1%	92.9%	(0.7%)	<0.01
	w_2	100.0%	99.7%	99.7%	93.9%	97.0%	(0.3%)	92.5%	95.3%	(0.5%)	<0.01
	w_3	100.0%	99.1%	99.1%	92.5%	97.6%	(0.3%)	92.1%	95.9%	(0.3%)	<0.01
	w_4	99.9%	99.8%	99.9%	93.5%	97.0%	(0.5%)	70.4%	93.8%	(1.2%)	<0.01
	w_5	99.8%	99.2%	97.8%	83.2%	90.7%	(1.0%)	80.1%	88.4%	(0.8%)	<0.01
	w_6	99.9%	99.0%	98.4%	73.2%	90.6%	(1.9%)	66.4%	83.3%	(3.2%)	<0.01
$n_\mathrm{m} = 700$	w_1	100.0%	99.2%	99.2%	93.2%	96.0%	(0.3%)	86.9%	92.8%	(0.5%)	<0.01
	w_2	100.0%	99.7%	99.7%	93.4%	96.8%	(0.2%)	91.2%	95.2%	(0.4%)	<0.01
	w_3	100.0%	99.1%	99.1%	94.2%	97.7%	(0.2%)	90.4%	95.8%	(0.3%)	<0.01
	w_4	99.9%	99.8%	99.9%	93.4%	97.1%	(0.4%)	73.1%	93.7%	(0.9%)	<0.01
	w_5	99.8%	99.2%	97.8%	82.6%	91.1%	(0.7%)	78.3%	88.4%	(1.1%)	<0.01
	w_6	99.9%	99.0%	98.4%	76.7%	91.2%	(2.1%)	68.6%	83.5%	(4.0%)	<0.01

this fact. A possible explanation for this could be that within our warehouse only 175 strips exist, such that only a few conflicts can be resolved when considering always the paper rolls at the front of each strip only. A limitation of GRP to at most 100 moves seems, however, reasonable for an application in our paper production company. When applying GRP on warehouse states obtained via our stocking strategy, the improvement potential is rather limited.

Regarding the performances of our VND variants it turned out that DRVND seems to be the best VND setting for relocations consisting of many paper roll movements. If the available time is rather limited the performances of RVND, OVND and ROVND are similar. RVND, however, is outperformed almost always by DRVND. To validate this hypothesis we performed Wilcoxon rank sum tests. The resulting p-values are for nearly all tested instances below 0.01, which indicates that the assumption is in most cases correct with an error probability of at most one percent. For those warehouse states obtained via our stocking strategy an improvement could still be achieved using the VND variants which implies that conflicts induced by improper production sequences have an impact on the stocking strategy. Anyhow, it is important to assign the rolls to good storage locations from the beginning on, since the results obtained by the new stocking strategy could not be reached by the relocation procedure applied to warehouse states resulting from the formerly used stocking strategy.

With respect to runtime, GRP is the fastest approach with runtimes of at most 5 seconds. DRVND, which is the slowest VND variant, needs for computing 700 paper roll movements about 3,5 minutes, which is reasonable according the the warehouse manager of our paper production company.

Finally, we investigated the number of times one paper roll is relocated during storage reassignments. We observed that even for the test runs including 700 moves, multiple moves of individual paper rolls seldom occur.

7 Conclusions and Future Work

Within this work, we presented a stocking and two relocation strategies for a storage location assignment problem arising in a paper production company, whereas the number of conflicts, i.e., additional reallocation moves of paper rolls, during shipment should be minimized. Each individual storage location of the underlying warehouse applies a last-in, first-out throughput strategy. Experimental tests as well as feedback given by the warehousemen and warehouse manager revealed that using the proposed stocking strategy the warehouse states could be significantly improved. In addition, the developed relocation strategy based on variable neighborhood descent can be used to reduce the likelihood of conflicts occurring during stock removal. Anyhow, these reallocation operations are only executed during idle times of warehousemen dedicated to shipment, why the reliable stocking strategy proposed in Section 4 is crucial for efficient warehouse operations.

Although the results obtained by our methods are promising and improved the warehouse management of our paper production company substantially, the further improvement potential is high. For example, future work should concentrate on the parallel optimization of the paper production process and storage management. The production sequence of paper rolls should be adapted such that the storage locations could be better utilized without raising conflicts during stock removals. For this purpose, a deeper investigation of the ordering process might be necessary such that the data administration is enhanced and shipping dates are more reliable in the future since the most conflicts emerge due to customers picking their orders too late (or too early).

References

1. 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) (2007) 481–501
2. Prandtstetter, M., Raidl, G.R., Misar, T.: A hybrid algorithm for computing tours in a spare parts warehouse. In Cotta, C., Cowling, P., eds.: Evolutionary Computation in Combinatorial Optimization – EvoCOP 2009. Volume 5482 of LNCS., Springer (2009) 25–36
3. Eggenhofer, O.: Optimales Lager-Layout: Kommissionierung, Material- und Verkehrsfluss im Fokus. Getränkegrosshandel (5) (2007) 30–34
4. Williams, B.D., Tokar, T.: A review of inventory management research in major logistics journals: Themes and future directions. The International Journal of Logistics Management **19**(2) (2008) 212–232
5. Thron, T., Nagy, G., Wassan, N.: Evaluating alternative supply chain structures for perishable products. The International Journal of Logistics Management **18**(3) (2007) 364–384
6. Albers, S.: Online algorithms: a survey. Mathematical Programming **97**(1) (2003) 3–26
7. Brynzér, H., Johansson, M.I.: Storage location assignment: Using the product structure to reduce order picking times. International Journal of Production Economics **46–47** (1996) 595–603 Proceedings of the 8th International Working Seminar on Production Economics.
8. Abdel-Hamid, A.A.A., Borndörfer, R.: On the complexity of storage assignment problems. Technical Report SC-94-14, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Berlin, Germany (1994)
9. Hassini, E., Vickson, R.G.: A two-carousel storage location problem. Computers & Operations Research **30**(4) (2003) 527–539
10. Jane, C.C.: Storage location assignment in a distribution center. International Journal of Physical Distribution & Logistics Management **30**(1) (2000) 55–71
11. Hansen, P., Mladenović, N.: Variable neighborhood search. In Glover, F.W., Kochenberger, G.A., eds.: Handbook of Metaheuristics. Kluwer Academic Publisher (2003) 145–184
12. Ritzinger, U.: Generierung von Ein- und Umlagervorschlägen in Lagern mit einer Last-In First-Out Strategie und kundenspezifischen Auslagerpräferenzen. Master's thesis, Vienna University of Technology, Austria (2008)
13. Hu, B., Raidl, G.R.: Variable neighborhood descent with self-adaptive neighborhood-ordering. In Cotta, C., et al., eds.: Proceedings of the 7th EU/MEeting on Adaptive, Self-Adaptive, and Multi-Level Metaheuristics. (2006)
14. Puchinger, J., Raidl, G.R.: Bringing order into the neighborhoods: Relaxation guided variable neighborhood search. Journal of Heuristics **14**(5) (2008) 457–472