European Journal of Operational Research 259 (2017) 293–307

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Innovative Applications of O.R.

Sequencing of picking orders in mobile rack warehouses

Nils Boysen[a,*], Dirk Briskorn[b], Simon Emde[c]

[a] Lehrstuhl für Operations Management, Friedrich-Schiller-Universität Jena, Carl-Zeiß-Straße 3, Jena 07743, Germany
[b] Schumpeter School of Business and Economics, Bergische Universität Wuppertal, Lehrstuhl für Betriebswirtschaftslehre insb. Produktion und Logistik, Rainer-Gruenter-Str. 21, Wuppertal 42119, Germany
[c] Fachgebiet Management Science/Operations Research, Technische Universität Darmstadt, Hochschulstraße 1, Darmstadt 64289, Germany

ARTICLE INFO

Article history:
Received 26 February 2016
Accepted 25 September 2016
Available online 28 September 2016

Keywords:
Warehousing
Mobile racks
Picker routing
Sequencing

ABSTRACT

A growing population and increasing real estate costs in many urbanized areas have made space for roomy warehouses with single-deep storage and wide aisles scarce and expensive. Mobile rack warehouses increase the space utilization by providing only a few open aisles at a time for accessing the racks. Whenever a stock keeping unit (SKU) is to be retrieved, neighboring racks mounted on rail tracks have to be moved aside by a strong engine, so that the adjacent aisle opens and the SKU can be accessed. As moving the heavy racks takes considerable time, the resulting waiting time determines large parts of the picking effort. It is, thus, advantageous to sequence picking orders, such that the last aisle visited for the preceding order is also the first aisle to enter when retrieving a subsequent picking order. We formalize the resulting picking order sequencing problem and present suited exact and heuristic solution procedures. These algorithms are tested in a comprehensive computational study and then applied to explore managerial aspects, such as the influence of the number of open aisles on the picking effort.

1. Introduction

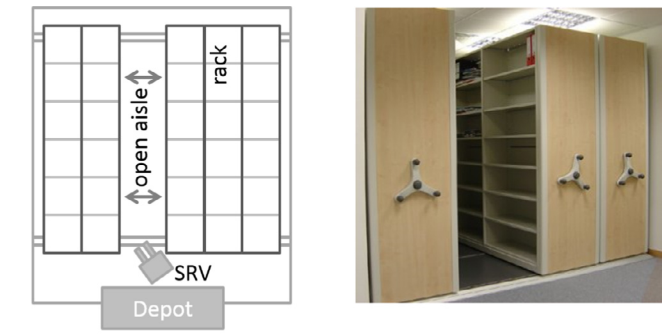
The population growth and a rural exodus in many parts of the world have made storage space scarce and expensive – especially in large urban areas where the lion's share of customers live. In some metropolises, e.g., in Singapore (see Zhao, Choa, & Broms, 1996), this development has even made underground warehouses a conceivable alternative. A more compact storage also decreases the energy consumption for heating, cooling, lighting and/or ventilating a warehouse, which – in times of increasing ecological awareness of customers and high energy prices – is a pressing concern in many supply chains. These are just two aspects that have made high density storage a focal topic among both researchers and practitioners alike in recent years. In this context, different compact storage systems, for instance puzzle-based storage (see Gue & Kim, 2007), the live-cube system (see Zaerpour, Yu, de Koster, 2016), or 3D ASRS (automated storage and retrieval system; see Yu & de Koster, 2009) have been investigated. This paper treats an alternative system based on mobile racks.

1.1. Order sequencing in a mobile rack system

In a mobile rack system (see Fig. 1), the parallel racks are densely packed and mounted on rails, so that the neighboring racks need to be moved aside to access a specific aisle. In the most compact form, the total width of all racks is only slightly shorter than the rail tracks, so that only a single aisle can be open at a time. Traditionally, mobile racks (also denoted as roller racks or mobile shelves) are moved manually, e.g., by turning a star handle, and applied for storing rarely accessed documents and books in archives and libraries. An example for these old-established systems is depicted in Fig. 1b. However, mobile rack systems can also be automated, which means that some strong engine is applied to move the racks once an escapement mechanism at the front of the racks, e.g., a button, ripcord, or light barrier, is activated. An open aisle can, then, be accessed by a human picker or some storage and retrieval vehicle (SRV) in order to pick the stock keeping units (SKUs) defined by a pick list. An SRV can either be a man-aboard system or an unmanned vehicle, e.g., with a robotic arm for automatically withdrawing items (see Chang, Fu, & Hu, 2007; Hu, Chang, Fu, & Yeh, 2009). Automated mobile racks are mainly applied in refrigerated warehouses, e.g., for frozen food or pharmaceuticals, where saving energy through compact storage is a pressing concern (SSI Schäfer, 2013). A case study for such a setting is reported in Section 1.2.

* Corresponding author.
E-mail addresses: nils.boysen@uni-jena.de (N. Boysen), briskorn@uni-wuppertal.de (D. Briskorn), emde@bwl.tu-darmstadt.de (S. Emde).
URL: http://www.om.uni-jena.de/ (N. Boysen), http://www.prodlog.uni-wuppertal.de/ (D. Briskorn), http://www.or.wi.tu-darmstadt.de/ (S. Emde)

http://dx.doi.org/10.1016/j.ejor.2016.09.046
0377-2217/



(a) schematic warehouse layout (b) mobile shelving in archives

Fig. 1. Mobile shelving[1].

Fully loaded racks may become very heavy, so that moving the racks and opening another aisle causes considerable waiting time for the SRV. In fact, idle time in front of an opening aisle is the main driver of the total picking effort, because the driving speed of SRVs is typically much higher than the velocity the racks are moved with. In a specific system implementation (see SSI Schäfer, 2013), for instance, the racks move only with 4 meters per minute compared to 115 meters per minute of the SRV. Weight limits restrict the maximum length of the racks to 45 meters, so that it takes an SRV at most 25 seconds to traverse an aisle. We, thus, approximate the total picking effort by only considering the number of aisle relocations when processing a given set of picking orders in a picker-to-parts, pick-by-order environment. In such a setting, it is advantageous to sequence the picking orders such that the last aisle visited for the preceding picking order is also the first aisle entered when processing the subsequent picking order. This way, the number of aisle relocations and, thus, the total picking effort is reduced.

The basic order sequencing and aisle relocation problem will be formally defined in Section 2.1. To support an intuitive understanding we briefly sketch this problem with the help of the following example.

Example 1. Consider a mobile rack system with three aisles, only one of which is open at a time. We have four picking orders to be successively processed under a pick-by-order policy. Each SKU has a single fixed shelf position in the warehouse, so that according to each pick list the aisles to be visited per order can be preprocessed. First order $o = 1$ requires the picker to visit aisles 1 and 2, while orders $o = 2$, $o = 3$, and $o = 4$ make it necessary to visit aisles 1 plus 3, only aisle 1, and aisle 1 plus 3, respectively. Initially, aisle 2 is open. For moving the racks and relocating the open aisle we presuppose two buttons installed on each rack, which move the rack and all adjacent predecessor racks into the selected direction. Fig. 2 depicts two alternative solutions, where the black arrows indicate the selected buttons to realize the depicted rack setting. Solution (a) entails three aisle relocations. In this solution, order picking is started with order $o = 1$, which requires some SKU from initially opened aisle 2. Then, the open aisle is relocated and aisle 1 is opened. Here, order $o = 1$ is completed and moved to the depot. Then, order $o = 3$ is processed, which can be fulfilled from already open aisle 1. After the return from the depot, order $o = 2$ is started, which also requires access to aisle 1. The order is completed after relocating the open aisle to aisle 3, where subsequently order $o = 4$ is started. This order is completed after the third aisle relocation to aisle 1. Alternative solution (b), where orders are processed in the sequence ⟨3, 1, 2, 4⟩, requires five aisle relocations and, thus, leads to a longer waiting time.

[1] Fig. 1b is published under the Creative Commons License and its author is Wikichops.

1.2. Case study

Order sequencing in mobile rack warehouses was brought to our attention by a large German retail chain. In a central distribution center, from which dozens of supermarkets are supplied, a large mobile rack system for frozen food is installed. During the day, the orders of the supermarkets are submitted to the distribution center, pallets of frozen food are assembled according to these orders, and intermediately stored in the refrigerated mobile rack system. In the morning of the next day, trucks arrive to load the pallets from the mobile rack system (and from other subsystems of the distribution center). Depending on the demand per supermarket, these trucks deliver the goods either directly to a single supermarket or multiple stores are successively visited in milk-run tours. Short-term changes require a frequent replanning of the truck tours, so that it is not always possible to store all pallets required by a specific truck in the same aisle of the mobile rack system.

In the morning, many trucks compete for the scarce loading capacity, so that delays due to an excessive number of tedious aisle relocations in the mobile rack system are not acceptable. In this setting, the pallets to be loaded on a truck represent a picking order and an efficient sequencing of these orders facilitates rapid truck loading processes.

The mobile rack system is separated into a few sections each operated by a single (un-manned) fully-automated SRV. As was already mentioned in Section 1.1, the velocity of the SRVs is high compared to the movement of the heavy racks, so that waiting for aisle relocation constitutes large parts of the total retrieval time. The single cross aisle at the front of the racks is equipped with a conveyor system, which transports the pallets towards the loading docks. Each SRV has to put pallets of the SKUs stored in their subsegment onto their dedicated conveyor segment, such that orders are not mixed. An aisle relocation is initiated by an electronic signal once the SRV arrives in front of a yet closed aisle. By reusing aisles between successive orders, the retrieval process in the morning can be accelerated, so that truck delays due to pallets arriving late from the mobile rack system become less likely.

Note that the need to quickly retrieve preassembled pallets to be loaded onto trucks is a quite common problem setting in the food industry. Zaerpour, Yu, and Koster (2015) and Boysen, Boywitz, and Weidinger (2016), for instance, report on a similar setting in a refrigerated compact warehouse. Both papers, however, treat the put-away process to avoid blockings in deep-lane storage systems.

Our order sequencing problem in mobile rack warehouses aims to reduce the problem structure to its very core. In many real-world applications of mobile racks further problem characteristics, e.g., multiple SRVs, release dates of picking orders, or alternative objective functions, e.g., including travel times, may need to be considered. In this paper, however, we aim the extract the core problem, which may be a valid building block in more general problem settings.

1.3. Literature review

Scientific research on optimizing warehouse operations has a long lasting tradition and a huge body of literature has accumulated over the years. Instead of trying to summarize all papers published on order picking, picker routing, and order sequencing we only refer to the excellent review papers published on these topics in recent years (see de Koster, Le-Duc, and Roodbergen, 2007; Gu, Goetschalckx, & McGinnis, 2007; 2010). In detail, we only survey those papers treating mobile racks.

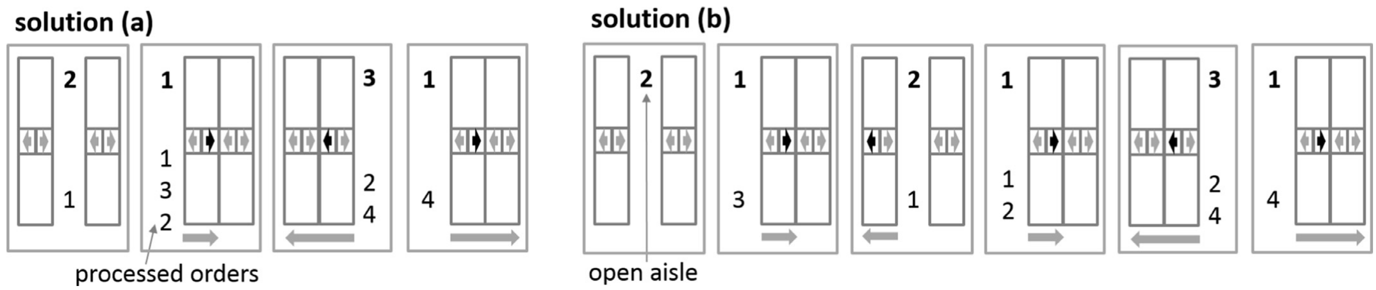


Fig. 2. Example 1 for the order sequencing and aisle relocation problem with a single open aisle.

In spite of their long tradition, just a few papers on mobile racks have been published. A reason for this might be the low performance gains promised by computerized decision support for manual mobile racks like they are often applied in libraries (see Russon, Kútik, & Clarke, 1982). However, in recent years automated mobile racks receive more attention and are mainly erected in refrigerated warehouses (see SSI Schäfer, 2013). To the best of the authors' knowledge the only papers having an operational research focus on automated mobile racks are the papers of Chang et al. (2007) and Hu et al. (2009). Both papers treat the picker routing problem for retrieving a single picking order. However, automated mobile rack warehouses typically consist of only a single cross aisle for entering and exiting each aisle from the front (also denoted as the "out-and-back" approach or the "return policy") in order to keep the space utilization as low as possible. In such a setting, picker routing leaves nearly no flexibility, i.e., each aisle from which an SKU is to be retrieved is to be opened and accessed. Therefore, both papers assume different additional characteristics resulting in non-trivial order picking problems. In Chang et al. (2007), the access of each aisle is restricted to the out-and-back policy, but each shelf can be accessed via both adjacent aisles. Thus, picking effort can be saved by a suited decision via which aisle each SKU is retrieved. In Hu et al. (2009), an additional middle-aisle (allowing separate open aisle positions before and after the middle aisle) and a two-sided shelf access on SKUs are considered. Both settings, however, are rather rare, so that we concentrate on the standard setting, i.e., no additional middle aisle and single-aisle shelf access (see SSI Schäfer, 2013). In this setting, order sequencing is the main concern.

It can be concluded that all former papers on mobile racks treat each picking order separately, so that to the best of our knowledge this paper is the first to investigate the picking order sequencing problem in a mobile rack warehouse.

From a technical perspective, our picking order sequencing problem with a single open aisle is a special case of the clustered traveling salesman problem (CTSP). In this problem, cities (SKUs) to be visited are clustered into subgroups (picking orders) and a minimal tour through all cities is sought where all cities belonging to the same cluster are to be visited in direct succession (e.g., see Chisman, 1975). However, as we will elaborate after providing a detailed problem description, our picking order sequencing problem has some special characteristics enabling tailored solution procedures not applicable to the CTSP.

Another stream of literature related to our problem is machine scheduling with setup times (e.g., see the survey paper of Allahverdi, Ng, Cheng, and Kovalyov, 2008 or the textbooks of Błażewicz, Ecker, Pesch, Schmidt, and Węglarz, 2007; Pinedo, 2012). On first sight, the need to relocate an aisle when changing between two successive picking orders (jobs) could be represented by sequence-dependent setup times. However, the decision whether or not the last aisle visited of a predecessor order can be reused by the subsequent one, does not exclusively depend on the order sequence, but also on the specific choice of each order's last and first aisle visited. Therefore, we have a more complicated problem structure and previous solution concepts are not directly applicable.

1.4. Contribution and paper structure

The aim of this paper is twofold. First, we formulate the basic order sequencing and aisle relocation problem and provide suited exact and heuristic solution procedures for the cases with a single and multiple open aisles. Furthermore, these algorithms are applied for exploring the basic trade-off between the aisle relocation effort and space utilization. Clearly, a mobile rack warehouse need not provide merely a single open aisle, but could also leave additional space for multiple aisles opened at the same time. In this case, the probability that aisles left open after processing a picking order can be "reused" by a successive picking order increases. On the other hand, any further open aisle requires additional space. In the extreme case, all aisles are opened in parallel, so that mobile racks become superfluous and a traditional single-deep warehouse with wide aisles arises. By exploring this basic trade-off warehouse managers receive some decision support when deciding on a proper layout, i.e., the number of open aisles, of a mobile rack warehouse.

The remainder of the paper is structured as follows. First, we formulate the basic order sequencing and aisle relocation problem and provide suited exact and heuristic solution procedures both if only a single aisle is available (Section 2) and if multiple open aisles exist (Section 3). Once these algorithms are at hand (and tested for their computational performance) we apply them in a comprehensive computational study to explore the basic trade-off between increasing space utilization and decreasing picking effort if the number of open aisles is increased and vice versa (Section 4). Finally, Section 5 concludes the paper.

2. Order sequencing with a single open aisle

This section is dedicated to mobile rack systems having only a single open aisle. We, first, give a detailed problem definition in Section 2.1. Then, in Sections 2.2 and 2.3 solution procedures are proposed. In Section 2.2 we focus on a decomposition with regard to the picking order sequence and the aisle relocation process while in Section 2.3 the problem is tackled in a holistic manner.

2.1. Problem description

The warehouse we consider consists of parallel, mobile racks where a set S of SKUs are stored. Each SKU is fixedly assigned to a single shelf space in one of the racks, so that a dedicated storage policy (see Bartholdi & Hackman, 2014) is applied. In order to access an SKU the neighboring aisle needs to be opened by moving the mobile racks aside, so that the SRV can enter the aisle and retrieve the number of items defined by the pick list. We denote the total set of aisles in the mobile rack system by $A = \{1, \dots, m\}$.

Table 1
Notation.

S	Set of SKUs
A	Set of aisles (with $A = \{1, \dots, m\}$)
O	Set of picking orders (with $O = \{1, \dots, n\}$)
$S_o \subseteq S$	Subset of SKUs demanded by order o
$A_o \subseteq A$	Subset of aisles to be visited for picking order o
a_s	Aisle from which SKU s is to be retrieved
δ	Aisle initially open
π	Sequence of triples decoding a solution (with $\pi = ((\pi_1^o, \pi_1^f, \pi_1^l), \dots, (\pi_n^o, \pi_n^f, \pi_n^l)))$
π_i^o	The ith picking order of sequence π
π_i^f	First aisle visited when picking the ith order of sequence π
π_i^l	Last aisle visited when picking the ith order of sequence π

For the moment we assume that there is only space for a single open aisle at a time. The alterations required if more than a single open aisle is available are elaborated in Section 3. Each SKU s is accessible via a specific aisle, denoted by a_s. At the beginning of the planning horizon aisle $\delta \in A$ is initially open. Whenever the mobile racks are moved and another aisle is opened we say the open aisle is *relocated*. The warehouse consists only of a single cross aisle at the front of the racks, so that an open aisle has to be entered and exited via the same opening, i.e., the out-and-back or return policy is applied. Note that this setting is typical for mobile rack warehouses (see Chang et al., 2007; Hu et al., 2009), because this way the most compact form of mobile rack storage can be realized.

In this setting, a given set $O = \{1, \dots, n\}$ of picking orders is to be processed. Each single picking order $o \in O$ consists of one or multiple order lines defining the set S_o of SKUs to be retrieved. Note that o can also represent a predefined batch of orders, if multiple picking orders are concurrently picked, i.e., a batching policy is applied (see de Koster et al., 2007). Then, our solution concepts can directly be applied to derive the sequence of batches. In the following, we will, nonetheless, only refer to picking orders. The retrieval process is organized according to the picker-to-parts, pick-by-order paradigm. This means that the (manned or unmanned) SRV successively moves to all SKUs defined by a specific picking order and, then, returns to the depot to unload the current items. Afterwards the next picking order is processed and successively picked until all picking orders are processed. We aim to explore the most basic problem setting, so that we consider only a single SRV and presuppose that neither precedence constraints among picking orders nor deadlines exist and assume all picking orders being available at the beginning of the planning horizon. As all items of an SKU are stored in their unique shelf space and, obviously, revisiting an aisle multiple times during the execution of the same picking order cannot be optimal, in our problem SKUs need not explicitly be differentiated and only the aisles to be visited per picking order have to be considered. Thus, $A_o = \bigcup_{s \in S_o} a_s$ defines the set of aisles to be visited for picking order o. The notation utilized in this section is summarized in Table 1.

For retrieving an item, the SRV has to move towards the respective aisle and can either directly enter (provided that the aisle is already open) or has to activate an escapement mechanism, e.g., a button, ripcord, or light barrier, at the front of the rack. Once the aisle relocation process is started, the SRV has to wait until the racks are moved and the respective aisle is open. In our problem setting we presuppose that the total picking effort is only influenced by the number of aisle relocations. On first sight, this seems an oversimplification of the real-world problem situation. However, the following arguments suggest that this is actually not the case:

- The time for withdrawing items from their shelf and later on processing them in the depot just depends on the type and amount of SKUs to be retrieved. Both is defined by the pick list and, thus, not influenced by the picking order sequence.
- The travel time of the SRV within an open aisle for accessing a specific shelf is also not influenced. Each SKU is only available in a single shelf, which is to be accessed from the front. Thus, the travel within the aisles is constant once the pick list is given.
- Finally, there remains the SRV travel from the depot to the first aisles, from aisle to aisle, and from the last aisle back to the depot. This part of the picking process, however, resembles some routing problem on a line, which typically has trivial solutions, e.g., go from the depot to the left most aisle and successively visit all other aisles from left to right and then return. Alterations of such a straightforward solution can only occur due to reusable open aisles. However, seeing the great differences in a typical SRV's velocity and the considerable time it takes for relocating an aisle (see Section 1), it seems pardonable to exclude these potential detours for the benefit for a more handy and compact problem formulation.

We, thus, assume that only the waiting time for rack movements is critical, so that in our setting the total time for retrieving picking order set O is only determined by the number of aisle relocations. Clearly, this number can be reduced if the last aisle visited for retrieving a predecessor picking order is also the first aisle to be entered when the subsequent picking order is picked. Therefore, what we are interested in throughout the remainder of the paper is the sequence of picking orders being processed and the first and the last, respectively, aisle visited while a picking order is processed. Note that this does not necessarily require determining the full sequence of aisles visited.

In this setting, we seek a permutation of all picking orders along with the first and last aisle visited per picking order, such that the total number of aisle relocations is minimized. More formally, we seek a sequence

$$\pi = ((\pi_1^o, \pi_1^f, \pi_1^l), \dots, (\pi_n^o, \pi_n^f, \pi_n^l)),$$

where π_i^o denotes the ith picking order of the sequence and π_i^f and π_i^l denote the first and the last aisles visited while processing this picking order π_i^o. For ease of notation let π_0^l represent the initially open aisle σ.

A feasible solution to our problem is a sequence π, such that

- $(\pi_1^o, \dots, \pi_n^o)$ contains each picking order in O exactly once,
- for each $(\pi_i^o, \pi_i^f, \pi_i^l)$ we have $\pi_i^f, \pi_i^l \in A_{\pi_i^o}$, that is the first and last aisle visited for retrieving picking order o must be in A_o, and
- $\pi_i^f \neq \pi_i^l$ or $|A_{\pi_i^o}| = 1$, that is the first and the last aisle can coincide only if it is the only aisle visited for retrieving the picking order.

Note that the last feasibility condition ensures that the aisle is not relocated to a position that has already been visited during the retrieval of the current picking order. Clearly, such a rack movement cannot reduce the total number of aisle relocations, because even if the revisited aisle can be reused by a successive picking order the very same movement could also be assigned to the successive picking order without altering the solution value. Adding this consideration to the problem definition, however, allows for a much easier calculation of the objective value. The number of aisle relocations according to feasible solution π can be derived as

$$Z(\pi) = \sum_{o \in O} |A_o| - \sum_{i=1}^{n} \Theta(\pi_i^f = \pi_{i-1}^l),$$

where $\Theta(x)$ is a function returning 1 if Boolean expression x is true and 0 otherwise. The picking order sequencing problem for a single open aisle (OS-single) is to find a solution π that minimizes the number $Z(\pi)$ of aisle relocations among all feasible solutions.

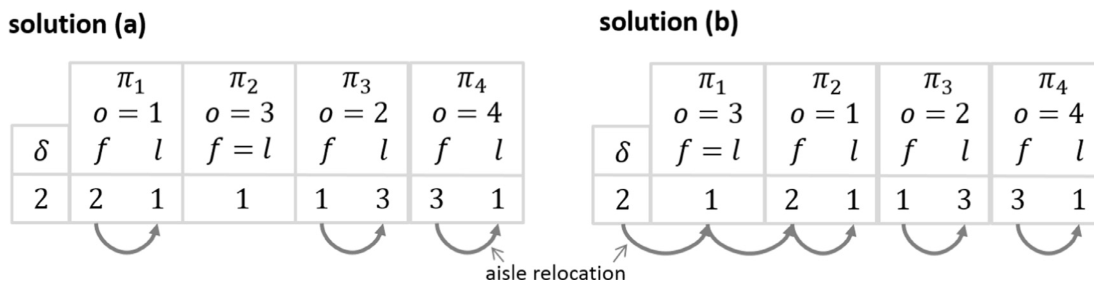


Fig. 3. Example 1 for OS-single.

Example 1 (cont.): To clarify our notation we restate Example 1 introduced in Fig. 2. We have a mobile rack warehouse with $m = 3$ aisles and $n = 4$ picking orders defined as follows: $A_1 = \{1, 2\}$, $A_2 = \{1, 3\}$, $A_3 = \{1\}$, and $A_4 = \{1, 3\}$. Initially, aisle $\delta = 2$ is open. Fig. 3 depicts the resulting solutions structures π for the two alternative solutions.

Unfortunately, solving OS-single turns out as a complex matter because of the following complexity status.

Theorem 1. *The OS-single problem is strongly NP-hard.*

Proof. See Appendix A. □

In the next two sections, exact and heuristic algorithms for solving OS-single are presented. Specifically, we introduce decomposition procedures, which separate order sequencing from aisle relocation (Section 2.2). Furthermore, dynamic programming (DP) based solution procedures directly tackling OS-single are presented in Section 2.3.

2.2. Heuristic decomposition procedures

In this section, we introduce heuristic solution procedures deciding on the sequence of picking orders and aisle relocation separately. Consequently, these approaches consist of two stages:

- The first stage considers the set of all picking order sequences as the search space and derives promising order sequences. For this purpose, we introduce a simple priority rule based heuristic and a simulated annealing (SA) procedure. Details are given in Section 2.2.1.
- Each generated sequence is evaluated in the second stage, where we minimize the aisle relocations for the given order sequence. This subproblem is shown to be efficiently solvable in polynomial time by a DP procedure described in Section 2.2.2. We develop a DP approach that finds the optimum among potentially $\mathcal{O}(m^n)$ solutions corresponding to a given sequence σ of picking orders in linear time. We denote by $\overline{Z}(\sigma)$ the minimum number of aisle relocations that can be achieved for a given order sequence σ.

These two stages are described in the following two sections.

2.2.1. Picking order sequences

In stage 1 of our decomposition, we, first, apply a straightforward priority rule based heuristic. It simply generates a single picking order sequence in a step-wise manner by successively fixing the next picking order. Specifically, picking order sequence σ is generated by selecting an order among the not yet selected ones for sequence positions $\sigma(i)$, with $i = 1, \dots, n$, which shares the most aisles with its preceding picking order. As a tie break rule, we select the order with the lowest index. We dub this approach the *most shared aisles rule*.

Example 1 (cont.): For our example of Fig. 3 the priority rule based heuristic generates picking order sequence $\sigma = \langle 1, 2, 4, 3]$, for which stage 2 of our decomposition approach determines $\overline{Z}(\sigma) = 3$ aisle relocations (see Section 2.2.2).

Furthermore, we apply an SA scheme (e.g., see Kirkpatrick, Gelatt, & Vecchi, 1983; Osman & Kelly, 2012) for generating picking order sequences in stage 1, outlined in Algorithm 1. SA is a well-known metaheuristic that hast been applied successfully to a wide range of difficult combinatorial optimization problems. It emulates the natural cooling process of a material previously heated to very high temperatures. In our case, a solution is directly encoded as the picking order sequence σ, which in each iteration is handed over to stage 2 of our decomposition approach in order to determine the objective value $\overline{Z}(\sigma)$.

Algorithm 1: Simulated annealing for solving OS-single.

```
σ := σ^best := random initial picking order sequence;
e := e^best := Z̄(σ);
i := 1;
while i ≤ 10,000 do
    τ := max_{o∈O}{|A_o|}
    while τ ≥ 0.1 and i ≤ 10,000 do
        for j := 1 to 100 do
            if rnd(0; 1) ≤ 0.1 then
                σ' := neighbor of σ (swap two random picking orders);
            else
                σ' := neighbor of σ (push one random picking order);
            e' := Z̄(σ');
            Δ := e' − e;
            if exp(−Δ/τ) > rnd(0; 1) then
                σ := σ';
                e := e';
            if e < e^best then
                σ^best := σ;
                e^best := e;
        τ := τ · 0.995;
        i := i + 1;
return σ^best;
```

Within SA, an initial solution is determined by a random picking order sequence. Starting from an initial temperature $\tau := \max_{o \in O}\{|A_o|\}$ and the first initial solution σ, for each temperature τ, an epoch of 100 iterations is started. In each iteration, a neighboring solution σ' of the current incumbent solution σ is reached by either swapping two picking orders assigned to two randomly selected sequence positions or by pushing one random picking order to another sequence position. The probability of a swap move is 10% and of a push move consequently 90%. If $\exp((\overline{Z}(\sigma) - \overline{Z}(\sigma'))/\tau)$ is greater than a uniformly distributed random number $0 \le rnd(0;\,1) \le 1$, the neighborhood solution replaces the current solution. After an epoch has ended, the temperature is lowered to $\tau := \tau \cdot 0.995$. If the temperature drops below 0.1, the procedure is restarted with a new initial solution. Finally, after the temperature has been lowered a total of 10,000 times, the optimization ends and the best solution found is returned.

Note that this procedure can be sped up in a multi-processor environment by running multiple copies of it concurrently, each starting from a different initial random solution, until the desired total number of 10,000 iterations is reached. Moreover, note that we only report the computational results for the above parameter

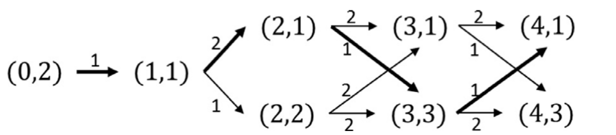


Fig. 4. DP graph for Example Example 1 and given picking order sequence $\sigma = \langle 3, 1, 2, 4]$.

setting of SA, because preliminary computational results indicated this setting as most promising.

In the next section, we elaborate how stage 2 of our decomposition approaches finds optimal aisle relocation plans for given order sequences.

2.2.2. Solving the aisle relocation subproblem

In this section, we treat the problem of finding the minimum number of aisle relocations for a given order sequence, i.e., calculating $\overline{Z}(\sigma)$. More formally speaking, for a given sequence σ of orders we look for an optimum solution $\pi = ((\pi_1^o, \pi_1^f, \pi_1^l), \ldots, (\pi_n^o, \pi_n^f, \pi_n^l))$ with $\pi_i^o = \sigma(i)$ for each $i = 1, \ldots, n$, where $\sigma(i)$ denotes the ith order in σ.

We can observe easily that a simple greedy approach (i.e., choose some last aisle that is shared by the successor) does not solve the problem to optimality. However, the problem can be solved in polynomial time. We first provide a straightforward DP approach, which runs in $\mathcal{O}(nm^2)$ time. Note that here we have an elementary distinction from the well-known CTSP (see Section 1.3). The CTSP remains strongly NP-hard even if the sequence of clusters is given. Seeing that our decomposition approach necessitates the evaluation of a lot of sequences, reducing the runtime seems desirable. We achieve this by exploiting special problem structures, which allow developing a more efficient DP procedure running in $\mathcal{O}(V)$ time, where V is the total number of aisle visits and is described in the following.

For ease of notation let $\sigma(0)$ be a dummy picking order $\bar{o}$ and let $A_{\bar{o}} = \{\delta\}$. We propose to employ states (i, a) with

- $i \in \{0, \ldots, n\}$ representing the first i picking orders in the sequence to be completed and
- $a \in A_{\sigma(i)}$ representing the aisle left open after finishing the ith picking order.

The minimum number of aisle relocations corresponding to state (i, a) is represented by $g(i, a)$. The initial state is $(0, \delta)$ with $g(0, \delta) = 0$.

A transition represents the processing of a single picking order. Hence, we consider a transition from state (i, a) to state $(i+1, a')$. Cost $c(i, a, a')$ of the transition from (i, a) to $(i+1, a')$ equals $|A_{\sigma(i+1)}| - 1$ if

- $a = a'$ and a' is the only aisle in $A_{\sigma(i+1)}$ or
- $a \in A_{\sigma(i+1)} \setminus \{a'\}$

and equals $|A_{\sigma(i+1)}|$ otherwise. This means that one aisle relocation is saved once a and a' being open after finishing i and $i+1$, respectively, allow to visit a as first aisle while processing $\sigma(i+1)$.

Now, we are equipped to formulate the Bellman function as

$$g(i, a) = \min\left\{g(i-1, a') + c(i-1, a', a) \mid a' \in A_{\sigma(i-1)}\right\}.$$

The optimum solution, then, is represented by $\min\{g(n, a') | a' \in A_{\sigma(n)}\}$. In this approach we have $\mathcal{O}(nm)$ states and $\mathcal{O}(nm^2)$ transitions and, therefore, the algorithm runs in $\mathcal{O}(nm^2)$ time.

Example 1 (cont.): Consider our example of Fig. 3 and a given picking order sequence of $\sigma = \langle 3, 1, 2, 4]$. The resulting DP graph is depicted in Fig. 4. The bold-faced optimal solution corresponds to solution (b) in Fig. 3.

In the following, we propose a more elaborate method finding the optimum solution in $\mathcal{O}(V)$ time. The structure is similar to the DP approach proposed above as we have $n+1$ layers of states. However, we will see that for each layer i we need at most three states which boils the number of transitions down to $\mathcal{O}(n)$. For ease of notation, let us reduce the costs of the transitions between layers i and $i+1$ to either 1 or 0 if an aisle relocation for the first aisle visited is required or not, respectively.

Let $S(i)$ be the set of states in layer i that are starting points of at least one transition of cost 0, that is the set of those states which have potential to save an aisle relocation when processing $\sigma(i+1)$. Let $\bar{S}(i)$ be the set of states in layer i but not in $S(i)$. Note that

1. state $(i, a) \in \bar{S}(i)$ is connected to each state in layer $i+1$ by a transition of cost 1,
2. state $(i, a) \in S(i)$ is connected to at least $|A_{\sigma(i+1)}| - 1$ states in layer $i+1$ by a transition of cost 0 (the remaining transition – if existing – is to $(i+1, a)$ and has cost of 1), and
3. if $|S(i)| \geq 2$, then there is at least one transition from layer i to each state in layer $i+1$ having cost 0.

First, we state two properties that follow from points 1. to 3. above and help to argue for the shrinking of the state space.

Property 1. *For each pair of states $(i-1, a)$ and $(i+1, a')$, there are $|\bar{S}(i)|$ pairs of transitions indirectly connecting $(i-1, a)$ to $(i+1, a')$ and passing through one of the states in $\bar{S}(i)$. The connection from $(i-1, a)$ to $(i+1, a')$ having minimum total cost has total cost of 1 if there is a transition of cost 0 from $(i-1, a)$ to a state in $\bar{S}(i)$ and has total cost of 2 otherwise.*

Property 2. *For each pair of states $(i-1, a)$ and $(i+1, a')$, there are $|S(i)|$ pairs of transitions indirectly connecting $(i-1, a)$ to $(i+1, a')$ and passing through one of the states in $S(i)$. If $|S(i)| \geq 3$ the connection from $(i-1, a)$ to $(i+1, a')$ having minimum total cost has total cost of 0 if there is a transition of cost 0 from $(i-1, a)$ to a state in $S(i)$ and has total cost of 1 otherwise.*

Based on these insights, we define a DP as follows. For each layer i, we consider a single state $(i, \bar{S}(i))$ if and only if $|\bar{S}(i)| \geq 1$ and

- state $(i, S(i))$ if $|S(i)| \geq 3$,
- states (i, a_1^i) and (i, a_2^i) if $S(i) = \{(i, a_1^i), (i, a_2^i)\}$, and
- state (i, a_1^i) if $S(i) = \{(i, a_1^i)\}$.

Hence, we represent all states in $\bar{S}(i)$ (in the first DP) by a single state. Furthermore, we represent all states in $S(i)$ (in the first DP) by a single state if there are at least three states in $S(i)$ and represent each state in $S(i)$ by an individual state otherwise.

We consider transitions as follows.

- We have a transition from $(i, \bar{S}(i))$ to each state in layer $i+1$ having cost 1 since each original state represented by $(i, \bar{S}(i))$ has a transition to each original state in layer $i+1$ of cost 1.
- We have a transition from $(i, S(i))$ to each state in layer $i+1$ having cost 0 due to Property 2.
- There is a transition from state (i, a_1^i) $((i, a_2^i))$ to state $(i+1, \bar{S}(i+1))$ if the original state (i, a_1^i) $((i, a_2^i))$ has at least one transition to one original state represented by $(i+1, \bar{S}(i+1))$. This transition has
    - cost 0 if the original state (i, a_1^i) $((i, a_2^i))$ has a transition of cost 0 to at least one original state represented by $(i+1, \bar{S}(i+1))$ (in particular, this is the case if $|\bar{S}(i+1)| \geq 2$) due to Property 1 and
    - cost 1 otherwise.
- From state (i, a_1^i) $((i, a_2^i))$ to state $(i+1, S(i+1))$ – if existing – there is a transition of cost 0 since there is a transition of cost 0

from original state (i, a_1^i) $((i, a_2^i))$ to at least two original states represented by $(i+1, \bar{S}(i+1))$ due to point 2. above (which means we can pass through one of them to each original state in layer $i+2$ with total cost of zero for both transitions).
- From state (i, a_1^i) $((i, a_2^i))$ to state (i, a_1^{i+1}) $((i, a_2^{i+1}))$ – if existing – there is a transition if there is a transition between these two original states. This transition's cost equals the cost of the corresponding original transition.

By careful implementation, we can design the state space and transition space in $\mathcal{O}(V)$ time.

We have $\mathcal{O}(n)$ states and $\mathcal{O}(n)$ transitions and, thus, once the state space and transition space is known we can find the minimum cost "path" through the state space in $\mathcal{O}(n)$ time. It remains to show how to extract the solution from the path through the condensed network.

In case we have $|\bar{S}(i)| = 1$ or $|S(i)| \leq 2$, the corresponding states $(i, \bar{S}(i))$ and (i, a_1^i) $((i, a_2^i))$ directly correspond to an original state, so there is no ambiguity here. Now consider the path starting at the initial state and the first state encountered which is either of form $(i, \bar{S}(i))$ with $|\bar{S}(i)| > 1$ or of form $(i, S(i))$ with $|S(i)| > 2$.

In case the first such state is of form $(i, \bar{S}(i))$, we choose an original state in $\bar{S}(i)$ which is connected to the distinctly identified predecessor state by a transition of the same cost as in the condensed network (there is one by construction). If there is more than one candidate we can choose one arbitrarily since they all have identical transition costs to states in layer $i+1$.

In case the first such state is of form $(i, S(i))$, our choice depends on the distinctly identified predecessor state and the transition used to layer $i+1$. We must choose the original state from $S(i)$ such that it allows a connection from the predecessor state to the state in layer $i+1$ with the same cost on both involved transitions. By construction of the network there is one due to Property 2.

These results can be summarized by the following theorem.

Theorem 2. *OS-single with a given picking order sequence is solvable in $\mathcal{O}(V)$.*

Example 1 (cont.): In our example, the DP graph for the second, more efficient DP has the same structure as the one depicted in Fig. 4, that is, for this instance we cannot reach a reduction of the state space. In Appendix B, we present a larger example to exemplify the benefits of the second DP.

With an exact solution procedure on hand which efficiently evaluates given order sequences, our decomposition approaches, i.e., a priority rule based approach and simulated annealing, can conveniently explore the search space defined by all potential order sequences.

In the next section, we derive some solutions procedures which do not decompose the problem into two separate stages but aim to directly solve OS-single.

2.3. Holistic solution procedures

In this section, we propose solution procedures directly tackling OS-single, that is, we determine the order sequence and the aisle relocations in an integrated manner. Again, the methods are based on a DP scheme. This one extends the basic DP for sequencing problems provided by Held and Karp (1962), so that picking orders are added one by one to a partial schedule constituting the first picking orders to be processed. Before defining the DP approach formally we first develop properties regarding the problem structure. These properties enable us to reduce the average run time behavior.

Consider an optimum solution π, the partial sequence

$$((\pi_1^o, \pi_1^f, \pi_1^l), \ldots, (\pi_k^o, \pi_k^f, \pi_k^l)),$$

with $k < n$, and picking order π_{k+1}^o, scheduled next in π.

Lemma 1. *There is an optimum solution where $\pi_{k+1}^f = \pi_k^l$ if $\pi_k^l \in A_{\pi_{k+1}^o}$.*

Proof. Consider an optimum solution

$$((\pi_1^o, \pi_1^f, \pi_1^l), \ldots, (\pi_k^o, \pi_k^f, \pi_k^l), (\pi_{k+1}^o, \pi_{k+1}^f, \pi_{k+1}^l),$$
$$(\pi_{k+2}^o, \pi_{k+2}^f, \pi_{k+2}^l), \ldots, (\pi_n^o, \pi_n^f, \pi_n^l))$$

with $\pi_k^l \in A_{\pi_{k+1}^o}$ and $\pi_{k+1}^f \neq \pi_k^l$. We can simply replace $(\pi_{k+1}^o, \pi_{k+1}^f, \pi_{k+1}^l)$ by $(\pi_{k+1}^o, \pi_k^l, \pi_{k+1}^f)$ and obtain a feasible solution without changing the number of relocations due to the following. First, we save a relocation of the open aisle between picking orders π_k^o and π_{k+1}^o and we can add at most one relocation between picking orders π_{k+1}^o and π_{k+2}^o (in case $k < n-1$). Hence, the number of relocations cannot be increased. Second, since π is assumed to be optimum the number of relocations cannot be decreased either. □

Lemma 1 enables us to choose the first aisle to be visited greedily when attaching a picking order to the partial sequence. If the last aisle of the previous picking order can be used again, we do not consider any alternatives. Now, let us consider two picking orders o and o' with $A_o \subseteq A_{o'}$.

Lemma 2. *There is an optimum solution where*

- *picking order o precedes another picking order o' with $A_o = A_{o'}$ and $o < o'$, and*
- *picking order o in position k of the sequence precedes another picking order o' with $A_{o'} \subsetneq A_o$ only if $\pi_{k-1}^l \in A_o \setminus A_{o'}$.*

We abstain from a formal proof since in both cases we can see easily that a switching argument concerning picking orders o and o' supports the lemma.

Lemma 3. *There is an optimum solution π with $\pi_k^l \neq \pi_{k+1}^f$ only if $\pi_k^l \notin \bigcup_{q=k+1}^n A_{\pi_q^o}$.*

Proof. Consider an optimum solution π with $\pi_k^l \neq \pi_{k+1}^f$ and $\pi_k^l \in A_{\pi_q^o}$ for some $q > k$ (if there is more than one q we randomly pick one of them). Note that we can see a solution as a sequence of subsequences of picking orders where no relocation appears in a subsequence and between each pair of consecutive subsequences there is a relocation. Thus, π_{k+1}^o is the first picking order in its subsequence.

1. If π_q^o is the last picking order or the first picking order in its subsequence, then we can simply detach it from its current subsequence and add it as last picking order to the subsequence of π_k^o.
2. If π_q^o is neither the last picking order nor the first picking order in its subsequence and $|A_{\pi_q^o}| = 1$, then we can simply remove it from its current subsequence and add it as last picking order to the subsequence of π_k^o. Note that no additional relocation is necessary between π_{q-1}^o and π_{q+1}^o since $\pi_{q-1}^l = \pi_{q+1}^f$.
3. If π_q^o is neither the last picking order nor the first picking order in its subsequence and $|A_{\pi_q^o}| > 1$, then we have $\pi_q^l \neq \pi_k^l$ or $\pi_q^f \neq \pi_k^l$. In the first case, we can append π_q^o and all picking orders in the same subsequence following π_q^o to the subsequence of π_k^o. We choose π_k^l to be the first aisle visited while processing π_q^o which is feasible since $\pi_k^l \in A_{\pi_q^o}$ and π_k^l is not visited last while processing π_q^o. In the second case, we can append π_q^o and all picking orders in the same subsequence preceding π_q^o to the subsequence of π_k^o in reverse order. We, again, choose π_k^l to

$$(\emptyset, 2) \xrightarrow{1} (\{1\}, 1) \xrightarrow{0} (\{1{,}3\}, 1) \xrightarrow{1} (\{1{,}2{,}3\}, 3) \xrightarrow{1} (\{1{,}2{,}3{,}4\}, 1)$$

Fig. 5. DP graph for Example 1.

be the first aisle visited while processing π_q^o which is feasible since $\pi_k^l \in A_{\pi_q^o}$ and π_k^l is not visited first while processing π_q^o (in the previous solution).

In each case, we do not increase the number of aisle relocations. By repeating this procedure we can obtain a solution as specified in the lemma. □

We are now equipped to formulate the DP approach. We propose to employ states (O', a), $a \in \bigcup_{o \in O'} A_o$, with

- $O' \subseteq O$ representing the picking orders already scheduled and
- $a \in A$ representing the aisle left open after finishing the last of these picking orders.

The minimum number of aisle relocations corresponding to state (O', a) is represented by $f(O', a)$. The initial state is $(\emptyset, \delta)$ with $f(\emptyset, \delta) = 0$.

A transition represents the processing of a single picking order. Hence, we consider a transition from state (O', a) to state (O'', a') if $O'' = O' \cup \{o\}$ with $o \notin O'$, $a' \in A_o$, and either $|A_o| = 1$ and $a = a'$ or $|A_o| > 1$ and $a \neq a'$. We do not consider those transitions violating the structure implied by Lemma 1, 2, or 3. Furthermore, if identical picking orders remain, we only develop the one having the lower index.

The cost $c(O', a, o, a')$ of such a transition either equals $|A_o|$ if $a' \notin A_o$ or equals $|A_o| - 1$ otherwise. This is in line with Lemma 1.

Finally, we formulate the Bellman function as

$$f(O', a) = \min \Big\{ f(O' \setminus \{o\}, a') + c(O', a, o, a') \mid \text{ the transition from } (O' \setminus \{o\}, a') \text{ to } (O', a) \text{ is considered} \Big\}.$$

We have $\mathcal{O}(m2^n)$ states and $\mathcal{O}(m^2 n 2^n)$ transitions and, thus, can determine the optimum solution in $\mathcal{O}(m^2 n 2^n)$ time.

Example 1 (cont.): For our example instance given in Fig. 3 our lemmas reduce the DP graph considerably (see Fig. 5). This solution equals the optimal solution (a) depicted in Fig. 3.

In line with our complexity result, the state space of our DP grows exponentially. Consequently, we apply our DP-scheme in a heuristic beam search approach. Beam search neglects all but the *BW* (beam width) most promising states per stage. A stage k here refers to the set of states where exactly k picking orders are already scheduled. While this makes traversing the state space much faster possible, it comes at the price of potentially missing the optimum. To judge whether a specific state (O', a) belongs to the *BW* most promising states of stage $k = |O'|$, we add a lower bound value

$$LB(O', a) = \sum_{o \in O \setminus O'} |A_o| - n + k \tag{1}$$

on the number of aisle relocations for processing the remaining picking order set to current value $f(O', a)$. This bound, which simply assumes that all remaining picking orders can reuse their initial aisle, can quickly be calculated in $\mathcal{O}(1)$ if we store the total number of aisles still to be visited with each state and stepwise decrease this value with any additional picking order.

Example 1 (cont.): For our example introduced in Fig. 3, the lower bound for the initial state and state ({1, 3}, 1) amounts to $LB(\emptyset, 2) = 3$ and $LB(\{1, 3\}, 1) = 2$, respectively.

3. Order sequencing for multiple open aisles

A natural generalization of OS-single treated in Section 2 is the multiple open aisles setting where the number of aisles that can be open at the same time is not restricted to one but to a given integer number $K \geq 1$. Additional open aisles increase the probability of an aisle reuse between successive picking orders, which promises fewer aisles relocations and, thus, a more efficient order picking process. On the other hand, the more open aisles are available the higher the space requirements for storing the same amount of items. Another disadvantage of multiple open aisles is the increasing intricacy of the aisle relocation process; multiple aisles cannot move independently of each other, which considerably complicates the picking order sequencing for multiple open aisles treated in this section. The interdependency between the current locations of the open aisles and their target positions can best be explained with the help of an example (see Fig. 6).

The initial situation is depicted in Fig. 6(a): the two open aisles are currently positioned at positions 2 and 3 and they are to be relocated to positions 3 and 5. Although the open aisle in position 3 is already properly located we require two aisle relocations, because an open aisle cannot “jump” over its neighboring open aisle. First, we have to relocate the open aisle from position 3 to 5 by pushing the button with the black arrow depicted in Fig. 6(b). Then, activating the black arrowed button relocates open aisle 2 to 3 and the intended aisle positions are realized. However, consider the initial situation of Fig. 6(a) and target positions 2 and 5 for the open aisles, so that again one aisle is properly located. Here, only a single aisle relocation is required (see Fig. 6(b)).

It, thus, depends on the current positions of the other open aisles how many relocation moves it takes to move a single open aisle from a given start position to a target position. This considerably complicates the picking order sequencing for multiple open aisles (denoted as OS-multiple). Consequently, we suggest a relatively simple heuristic, which decomposes a problem with $K > 1$ open aisles into K instances of OS-single by fixing $K - 1$ racks that are no longer allowed to be moved. Racks are fixed such that there is exactly one open aisle in between each pair of consecutive fixed racks (or a fixed rack and the virtual start or end aisle). For our above example (see Fig. 6) with $K = 2$ we could, for instance, fix rack 3 and assign one open aisle to the resulting warehouse area to the left consisting of aisles 1–3 and another open aisle to the right-hand area with aisles 4–6. Only the black buttons depicted in Fig. 7 are then allowed to be pushed because otherwise the fixed rack would move.

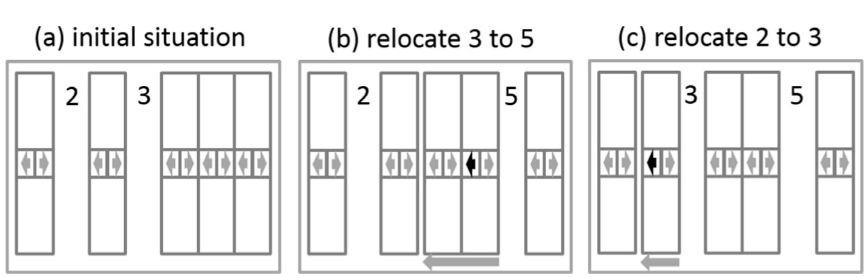


Fig. 6. Aisle relocation with multiple open aisles.

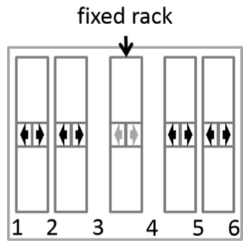


Fig. 7. Decomposition of OS-multiple to K OS-single problems by fixing racks.

Fixing $K-1$ racks as described above leads to $k=1,\dots,K$ disjunct warehouse areas, each servicing a set $\bar{A}_k \subset A$ of consecutive aisles and having a single open aisle. For generating the K instances of OS-single, each picking order in O has to be reduced to those aisles $A_o \cap \bar{A}_k$ falling in the respective area. Let $\bar{O}_k$ define such a reduced picking order set. Then the total objective value of OS-multiple amounts to the sum of all objective values for the K resulting instances of OS-single for $\bar{O}_k$ and $\bar{A}_k$ plus the aisle relocations required for moving an open aisle into each area if the initial positions of the open aisles deviate. Note that the K instances of OS-single cannot be solved completely independently of each other since the picking order sequence for each subproblem must be identical; otherwise, the picker may have to interrupt processing a picking order, which is not allowed in a pick-by-order setting. Hence, only the solution procedures described in Section 2.2 are applicable, where a global sequence is specified for all K subproblems either by priority rule or simulated annealing, and the actual aisle relocations are then optimized for each subproblem for the given sequence.

Given this simple decomposition of OS-multiple, the only question remaining is how to derive the disjunct warehouse areas, i.e., which racks to fix. For this problem, we presuppose two alternative solutions. The first, obvious approach is to simply partition the warehouse into equally sized areas with respect to the number of aisles served per area (as far as rounding differences permit).

A second approach determines the area borders, such that each warehouse area serves (nearly) an equal amount of aisle visits during the planning horizon. We, thus, aim to minimize the maximum number of aisles visits defined by the given picking order set falling into the respective area. Note that in our computational study we test whether this surrogate objective is indeed well-suited for increasing the number of aisle reuses. Let W_a denote the total number of picking orders that require access to aisle a. Then we aim at a solution vector $X=(0,x_1,\dots,x_{K-1},m)$ where x_k defines the last aisle accessible within area k, such that

$$\Gamma(X)=\max_{k=1,\dots,K}\left\{\sum_{a=x_{k-1}+1}^{x_k} W_a\right\}$$

is minimized subject to $x_k \ge x_{k-1}+1$ for each $k=1,\dots,K$. This problem can be solved in polynomial time by a straightforward DP approach, which is often applied to partition picking order sets into disjunct areas (e.g., Boysen & Fliedner, 2010). This DP employs states $(k,\, a)$ defining the last aisle $a \in A$ accessible by the one open aisle assigned to area $k \in \{1,\dots,K\}$. The minimum maximum workload associated with a state $(k,\, a)$ is denoted by $\gamma(k,\, a)$. We initialize the DP by introducing all states representing the first area $\gamma(1,a)=\sum_{\lambda=1}^{a} W_\lambda$ for each $a=1,\dots,m-K+1$. A transition from state $(k-1,a)$ to state $(k,\, a')$ with $a' > a$ represents the kth area ranging from aisle $a+1$ to a'. The minimum workload for such a transition can be determined via the Bellman recursion

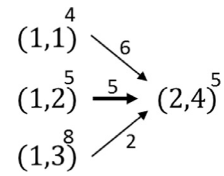


Fig. 8. DP graph for Example 2.

$$h(k,a)=\min_{k-1\le\lambda<a}\left\{\max\left\{h(k-1,\lambda);\ \sum_{\lambda'=\lambda+1}^{a} W_{\lambda'}\right\}\right\},$$

$\forall\,(k=2,\dots,K-1$ and
$a=k,\dots,k+m-K)$ or
$(k=K$ and $a=m)$.

The optimum solution can be derived by a simple backward recovery from final state $(K,\, m)$. We have $\mathcal{O}(Km)$ states and $\mathcal{O}(Km^2)$ transitions, so that the runtime of DP is in $\mathcal{O}(Km^2)$.

Example 2. Consider a warehouse with $K=2$ open aisles, where the $m=4$ aisles which are to be visited as follows: $W_1=4$, $W_2=1$, $W_3=3$, and $W_4=2$. The resulting DP graph and the bold faced optimal solution, which is to fix rack 2, is depicted in Fig. 8. The workload associated with a state can be found written next to the corresponding node. The workload assigned to the next area can be found next to the corresponding arc.

4. Computational study

In this section, we investigate the efficacy of the proposed algorithms by having them solve a set of test instances. Apart from the sheer computational performance, we test a rule-of-thumb commonly encountered in practice and review its quality. Moreover, we also investigate whether having multiple open aisles actually improves system performance to such a degree that it is worth the loss of storage density.

4.1. Instance generation

Since there are no established test data for the picking order sequencing problem, we will first describe how we generated the instances.

An instance consists of m aisles and a picking order set O containing $n=|O|$ picking orders. Each picking order $o \in O$, in turn, requires access to aisles A_o, the number of which we set to $|A_o| := rnd^{uni}(1,\dots,10)$, $\forall o \in O$, where rnd^{uni} is a randomly distributed integer (uniform distribution) from the interval in the argument. In typical warehouses, individual picking orders are not very diverse, meaning that a relatively low number of SKUs is responsible for a great part of the total flow through the warehouse (Bartholdi & Hackman, 2014). To account for the fact that some aisles are more popular than others, we first randomize the indices of the aisles, i.e., $\xi(i) := rnd^{uni}(1,\dots,m)$, $\forall i=1,\dots,m$, where $\xi(i) \neq \xi(i')$ iff $i \neq i'$. Then, we set $a := \xi(\max\{1;\ \min\{\lfloor rnd^{exp}(2.5)\rceil;\ m\}\})$, $\forall a \in A_o$ and $\forall o \in O$, where $\lfloor rnd^{exp}(2.5)\rceil$ is a randomly distributed number (power law distribution with exponent 2.5), rounded to the next integer. This makes it so that some (randomly selected) aisles (those with low indices ξ) are more likely to be accessed than others, which is in line with the situation in many real-world warehouses. The initially open aisle is set to $\delta=\lceil m/5\rceil$.

In order to account for different industries and warehouses of varying size, we varied the aisle count such that $m \in \{10, 20, 50\}$ and the picking order count such that $n \in \{25, 100\}$. For each combination of aisle and picking order count, we generated 20 instances, leading to a total of 120 data sets.

Table 2
Algorithmic performance for the instances that could be solved to optimality ($m = 10,\ n = 25$).

No.	DP	Gap (FCFS) (%)	Gap (MSR) (%)	Gap (BS-low) (%)	Gap (BS-high) (%)	Gap (SA) (%)	CPU s. (DP)	CPU s. (SA)
1	63	12.70	3.17	0.00	0.00	0.00	660.78	8.41
2	70	5.71	2.86	0.00	0.00	0.00	209.97	8.23
3	56	14.29	5.36	0.00	0.00	0.00	145.88	8.19
4	68	8.82	1.47	0.00	0.00	0.00	1222.61	8.96
5	61	14.75	6.56	0.00	0.00	0.00	351.98	7.37
6	47	17.02	4.26	0.00	0.00	0.00	25.34	7.14
7	56	17.86	3.57	0.00	0.00	0.00	366.68	7.85
8	64	10.94	3.13	1.56	0.00	0.00	136.19	7.83
9	76	7.89	2.63	0.00	0.00	0.00	1291.66	7.81
10	75	2.67	2.67	0.00	0.00	0.00	1685.81	9.06
11	62	14.52	6.45	0.00	0.00	0.00	414.56	7.62
12	63	9.52	4.76	0.00	0.00	0.00	369.00	7.64
13	52	21.15	3.85	1.92	0.00	0.00	120.05	7.18
14	67	10.45	8.96	1.49	1.49	0.00	994.30	7.94
15	59	11.86	1.69	0.00	0.00	0.00	387.17	8.34
16	57	14.04	3.51	0.00	1.75	0.00	868.79	8.69
Mean	62.25	12.14	4.06	0.31	0.20	0.00	578.17	8.02
Std. dev.	7.86	4.69	1.97	0.67	0.56	0.00	493.41	0.58
Std. error	1.97	1.17	0.49	0.17	0.14	0.00	123.35	0.15

4.2. Computational results

We implemented all algorithms presented in this paper in C# 5.0 and had them solve our test instances on an x64 PC equipped with an Intel Core i7-4930K 3.4 gigahertz CPU and 16 gigabytes of RAM.

First, we investigate the algorithmic performance for OS-single, where only $K = 1$ aisle may be open at a time. We propose an exact DP procedure to solve this problem in Section 2.3. Due to the NP-hard nature of OS-single, not all instances can be solved within a reasonable time frame by this approach, however. To be precise, we set a fixed time limit of 30 minutes and a memory limit of 8 gigabytes (half of the computer's physical RAM) for each instance. DP could solve only 16 of the small instances ($m = 10$ aisles and $n = 25$ picking orders) without exceeding either the time or the memory limit.

Table 2 lists the absolute optimal objective value (column *DP*) and the relative optimality gaps of five heuristics for OS-single for the 16 instances that could actually be solved by DP. Regarding the five heuristics, first, we turned DP into a beam search heuristic with small beam width $BW = 10$ (denoted *BS-low* in the table) and with large beam width $BW = 25$ (*BS-high*). Moreover, we used the decomposition heuristics from Section 2.2 to split OS-single into two distinct subproblems: One, determine an order sequence, and, two, decide on the first and last aisle to be visited. Accordingly, the three tested decomposition approaches are:

First come, first served (FCFS) Picking orders are processed in the order of their arrival (i.e., essentially randomly with respect to aisle relocation), and, in the second stage, the first and last open aisle are then determined greedily, meaning that the last open aisle of the ith picking order in the sequence is simply set to any random aisle that is shared with the $(i+1)$th picking order, if possible. Note that this way of scheduling picking orders and aisle relocations is obviously not particularly sophisticated; however, it reflects simple rules-of-thumb often observed in practice and can thus be used as a status-quo solution method for reference.

Most shared aisles rule (MSR) The picking order sequence is constructed by iteratively appending the picking order that shares the most common aisles with the current picking order, as described in Section 2.2. The open aisles are then determined optimally for the given sequence via the DP from the same section.

Simulated annealing (SA) This corresponds to the SA defined in Section 2.2, using DP to schedule the aisle movements for each given sequence.

As to the numerical results, first, the table reveals that the DP-based beam search presented in this paper is very capable of solving OS-single to (near-)optimality. While BS is somewhat sensitive to the beam width and struggles in some instances when $BW = 10$ (*BS-low*), it could solve almost all test instances with $m = 10$ aisles and $n = 25$ picking orders to proven optimality when $BW = 25$ (*BS-high*). For these small instances, the runtimes of BS as well as the priority rules were negligible across the board, coming to well less than 0.1 seconds, and are thus omitted from the table. SA invariably found the optimal solution for all instances, hence proving that coupling a metaheuristic for picking order sequencing and an exact DP for determining the open aisles is an effective approach for solving OS-single. The runtimes, listed in column *CPU s. (SA)* in seconds, however, are somewhat longer than those of BS, although still acceptable at significantly less than 10 seconds in all cases. In contrast, the priority rule-based procedures clearly deliver sub-optimal solutions. While MSR results may still be acceptable for practical purposes, especially in light of the great speed at which they are obtained, the very simple FCFS rule, although wide-spread in practice, produces substantial optimality gaps, exceeding 20% in one instance. Note that beside the mean values, the table also contains information on the standard deviation and the standard error of the mean.

While DP may be sufficiently fast for small warehouses or archives with few aisles and short-term planning horizons (maybe one hour), for larger problems heuristics are necessary. Therefore, we tested our procedures on the large instances where $m > 10$ and $n > 25$, too. Table 3 shows the lower bound $LB(\emptyset, \delta)$ on the objective value according to Eq. (1) in row *LB*, averaged over all 20 instances per parameter constellation; the rows labeled *gap* list the average relative deviation from that value for each algorithm. The CPU times for the three SA and BS variants can be found in the last three rows; the time taken to compute the priority rules was negligible (< 0.1 seconds for all instances) and is thus not in the table.

For the large instances, as for the small instances, the more sophisticated heuristics (BS and SA) clearly significantly improve on the results obtained by the priority rules. Especially when compared to the simplistic FCFS rule, the use of more powerful procedures can easily reduce the number of required rack moves between picking orders by at the very least 10%, in most cases

Table 3
Average performance of the heuristics.

	m	10	10	20	20	50	50
	n	25	100	25	100	25	100
LB	Mean	63.65	256.75	78.10	304.20	92.10	351.45
	Std. dev.	7.96	15.13	10.42	16.74	14.16	18.11
	Std. error	1.78	3.38	2.33	3.74	3.17	4.05
FCFS	Mean gap (%)	11.83	10.45	11.92	13.05	14.85	16.10
	Std. dev. (%)	4.87	2.24	3.99	2.02	4.76	1.99
	Std. error (%)	1.09	0.50	0.89	0.45	1.06	0.44
MSR	Mean gap (%)	3.87	2.03	3.69	2.84	4.70	3.17
	Std. dev. (%)	2.25	0.68	2.08	0.81	2.12	0.43
	Std. error (%)	0.50	0.15	0.47	0.18	0.47	0.10
BS-low	Mean gap (%)	0.44	0.00	1.10	0.35	4.07	1.11
	Std. dev. (%)	0.79	0.00	1.10	0.31	1.96	0.58
	Std. error (%)	0.18	0.00	0.25	0.07	0.44	0.13
BS-high	Mean gap (%)	0.43	0.02	0.76	0.17	2.94	0.64
	Std. dev. (%)	0.78	0.09	0.96	0.26	1.72	0.47
	Std. error (%)	0.17	0.02	0.21	0.06	0.39	0.10
SA	Mean gap (%)	0.19	0.00	0.08	0.02	1.00	0.30
	Std. dev. (%)	0.60	0.00	0.37	0.08	0.92	0.31
	Std. error (%)	0.13	0.00	0.08	0.02	0.21	0.07
CPU s. (BS-low)	Mean	6.46	0.34	0.02	0.36	0.02	0.31
	Std. dev.	27.38	0.02	0.00	0.03	0.00	0.03
	Std. error	6.12	0.01	0.00	0.01	0.00	0.01
CPU s. (BS-high)	Mean	0.03	0.88	0.04	1.15	0.04	0.98
	Std. dev.	0.01	0.05	0.01	0.09	0.00	0.07
	Std. error	0.00	0.01	0.00	0.02	0.00	0.02
CPU s. (SA)	Mean	8.23	36.27	8.32	36.89	8.12	35.00
	Std. dev.	0.70	1.51	0.70	1.69	0.89	1.36
	Std. error	0.16	0.34	0.16	0.38	0.20	0.31

Table 4
Average performance of the heuristics for $K = 2$ open aisles.

	m	10	10	20	20	50	50
	n	25	100	25	100	25	100
best obj. val.	Mean	44.10	180.70	60.25	233.30	79.55	298.35
	Std. dev.	6.73	12.02	8.54	15.36	11.37	15.25
	Std. error	1.50	2.69	1.91	3.43	2.54	3.41
FCFS	Mean gap (%)	42.85	39.80	34.66	37.78	28.45	32.29
	Std. dev. (%)	12.66	4.75	7.14	4.26	5.24	2.30
	Std. error (%)	2.83	1.06	1.60	0.95	1.17	0.51
MSR	Mean gap (%)	21.55	12.08	17.53	13.48	12.47	9.68
	Std. dev. (%)	5.66	2.72	4.85	2.66	3.73	1.42
	Std. error (%)	1.26	0.61	1.08	0.59	0.83	0.32
EQ-SA	Mean gap (%)	0.84	1.11	0.81	1.10	0.99	0.43
	Std. dev. (%)	1.61	1.95	1.40	1.57	1.55	0.48
	Std. error (%)	0.36	0.44	0.31	0.35	0.35	0.11
DP-SA	Mean gap (%)	0.81	0.45	0.52	0.12	0.55	0.19
	Std. dev. (%)	1.90	1.56	0.98	0.52	1.10	0.31
	Std. error (%)	0.42	0.35	0.22	0.12	0.25	0.07
CPU s. EQ-SA	Mean	14.86	59.16	14.85	58.96	13.65	57.20
	Std. dev.	1.76	3.68	1.71	4.54	1.35	3.78
	Std. error	0.39	0.82	0.38	1.01	0.30	0.84
CPU s. DP-SA	Mean	14.19	59.67	14.33	60.96	14.28	56.89
	Std. dev.	1.53	3.94	1.69	4.90	1.81	4.12
	Std. error	0.34	0.88	0.38	1.10	0.40	0.92

substantially more. Even employing a slightly more advanced rule, like MSR, can already improve results considerably, which should be helpful in time-critical situations. If CPU times are not of primary concern, BS and especially SA both deliver very good results hardly above the lower bound, improving upon MSR by about another 3–4% on average; however, BS tends to be somewhat faster than SA.

So far, we have only investigated OS-single with but one open aisle; however, some warehouses in fact allow more than one aisle to be open at a time. The decomposition-based heuristics, namely SA, MSR and FCFS, also work for the multiple open aisle case OS-multiple ($K > 1$). In order to use these solution procedures, the warehouse must first be split into disjunct areas, which can be done either by dividing the racks equally or via DP, as described in Section 3. Then, a sequence is determined either via SA or one of the priority rules, and, finally, K separate OS-single instances are solved. Care must be taken that each disjunct area actually contains an open aisle; we set the kth initial open aisle δ_k to either $k/5 \cdot m$ (if this aisle is actually within the kth area) or the first (i.e., leftmost) aisle within the kth area (else).

Table 4 contains the performance data of our proposed heuristics for OS-multiple with $K = 2$ open aisles: *best obj. val.* lists the average best objective value found by any of the heuristics, while the rows labeled *gap* show the average relative deviation from that value. The general picture is similar to the OS-single case: The priority rule-based heuristics perform comparatively poorly, on average about 36% (FCFS) and 14% (MSR) worse than the metaheuristic-based approaches. Regarding the latter, we tested SA with two settings: Either the warehouse was partitioned by simply assigning an (about) equal number of racks to each segment (denoted *EQ-SA*). Or we used the DP based method described in Section 3 to divide the racks between areas (denoted *DP-SA*). Note that the SA itself is otherwise identical in both cases.

SA performs very well regardless of how the warehouse is subdivided, both in terms of solution quality as well as CPU time, taking somewhere in the vicinity of one minute for the very largest instances. However, the tests show that there is a slight but definite difference between the two division methods. In all test settings, the more intelligent DP division method yields slightly improved results (about 0.5% fewer relocations on average), indicating that a suitable partitioning of the warehouse is in fact an important (albeit not critical) decision problem that can be solved adequately by our proposed DP.

To further investigate the practical implications of solving the picking order sequencing problem and to see just how much of a difference (near-)optimal schedules can make in practice, we propose to evaluate solutions the following way. For each aisle that has to be visited in a given solution, we assume the picking takes either 0.5 or 1.5 minutes on average (depending on the speed of the picker/the SRV, see Section 1). Each relocation move takes either one or three minutes (depending on the speed of the engines moving the racks). Given these parameters, we calculate both the average time it takes to fulfill a picking order if an FCFS schedule is used and if a near-optimal SA schedule is used (based on the data for $m = 50$ aisles, $n = 100$ picking orders and $K = 2$ open aisles). Accordingly, Fig. 9 shows how many picking orders can be expected to be fulfilled on average during an 8-hour shift.

Whether or not sophisticated scheduling techniques have a substantial impact on warehouse operations depends on the given equipment. The greater the speed difference between SRV and rack movements, the wider the gap between “good” and “bad” schedules. In some specific implementations (see Section 1), SRVs move at a speed in excess of 115 meters per minute whereas the racks creep along at only 4 meters per minute. In such a setting, we can expect the reduction in the number of relocations obtainable by SA to make a fairly large difference, as indeed it does, notable in the figure in the columns where the pick time per aisle is only one minute (that is, the SRV is fast). In these scenarios, an additional 6 or 7 picking orders can be filled during a shift. On first sight, this may seem unremarkable, but recall that the picking orders generated in our test instances visit 5.5 aisles on average, so that each of them takes about ten minutes processing time. Thus, a complete working hour can be saved and it can be expected that the gap would be even wider if the individual picking orders were smaller. In this case, the aisle relocations between picking orders (or lack thereof) would have relatively greater weight.

In the final part of our computational study, we investigate the benefit of allowing more than one aisle to be open at a time from a managerial perspective. On the one hand, the more open aisles

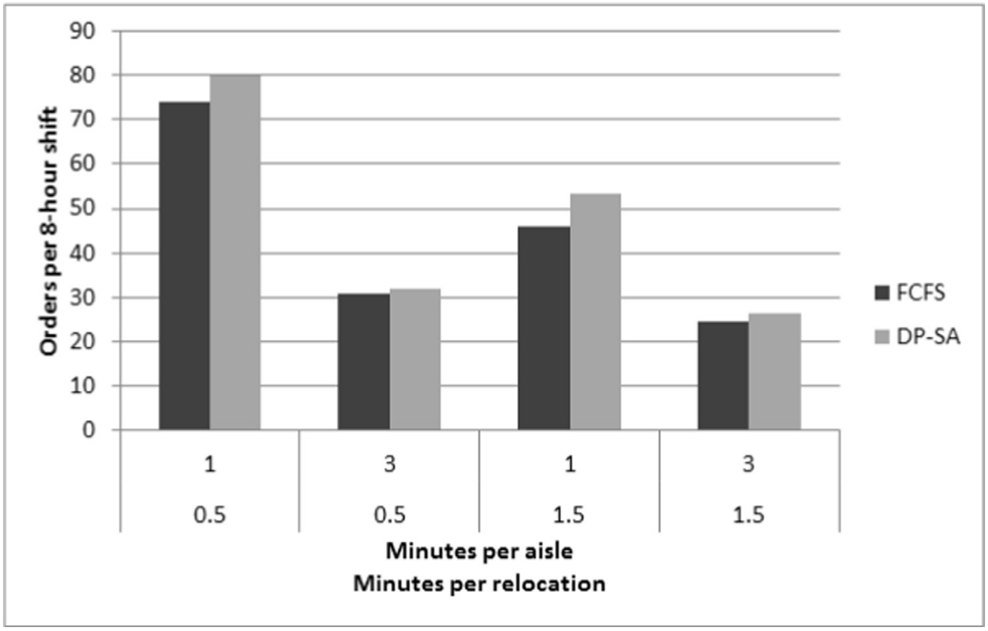
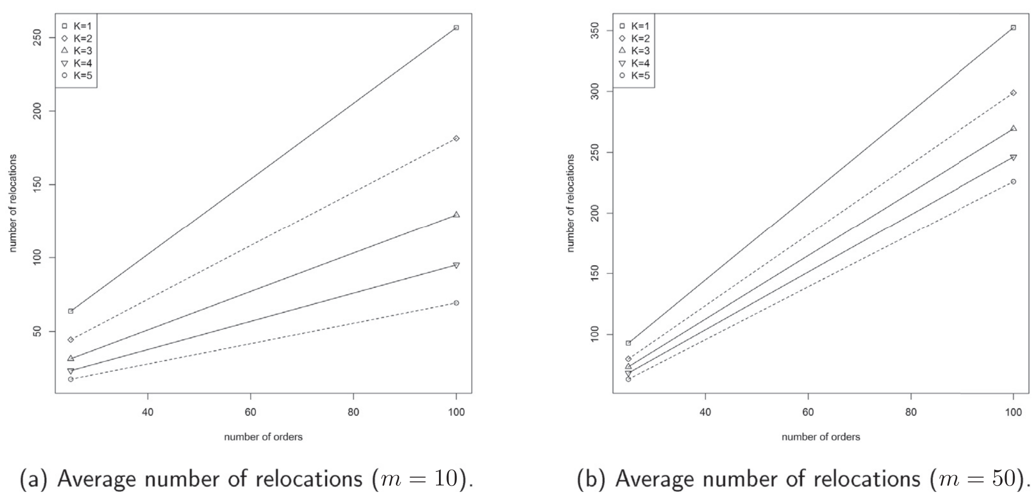


Fig. 9. Number of picking orders fulfilled per shift if SA instead of FCFS is used ($m = 50$, $K = 2$).



(a) Average number of relocations ($m = 10$).

(b) Average number of relocations ($m = 50$).

Fig. 10. Effect of opening additional aisles.

there are, the fewer relocations will be necessary and the quicker the picking. On the other hand, more open aisles also entail greater space requirements, partially negating one of the main advantages of mobile rack storage, and reduce energy efficiency in case of refrigerated warehouses. This begs the question whether it is efficient to open more than one aisle at all. In order to lend some decision support and quantify the gain in speed attainable by opening more than one aisle, we plot the number of relocations for $K = 2$, 3, 4, and 5 open aisles against the classic OS-single case ($K = 1$) in Fig. 10 (all results were gathered by DP-SA).

The data suggests that opening an additional aisle ($K = 2$) may be a worthwhile proposition in some cases, the average relative reduction of relocations coming in at about 30% for small warehouses ($m = 10$) and 15% for large warehouses ($m = 50$). The marginal benefit of opening even more aisles is diminishing, however; opening five instead of four aisles only reduces the objective value by about 26% and 8%, respectively. Considering that reducing the number of aisles, and hence the space and energy requirements, is the very purpose of installing mobile racks in the first place, these savings in relocations may not necessarily seem very impressive. To put this in perspective, if we assume that one aisle is about as wide as a rack, increasing the number of aisles by one would enlarge the warehouse area by about $(13 - 12)/12 \approx 8\%$ ($m = 10$) and $(53 - 52)/52 \approx 2\%$ ($m = 50$), respectively. Whether or not the faster pick time is worth it depends on the specific cost structure of the given warehouse.

5. Conclusion

This paper investigates the sequencing of picking orders in a mobile rack warehouse. Here, high-density storage is enabled by the racks being mounted on rails, on which neighboring racks are moved aside whenever a specific shelf is to be accessed. As racks are heavy and moving them (even in an automated setting) takes considerable time, we aim at an picking order sequence where aisles left open after processing a preceding picking order can be reused, i.e., will be accessed by the succeeding picking order as well. We formalize the resulting optimization problem and suggest suitable solution procedures for the settings where either only a single aisle is open at a time or multiple aisles are simultaneously accessible. From a managerial perspective, our computational results reveal the following two key findings:

- Computational picking order sequencing is especially valuable if the retrieval process inside the aisles is fast in relation to the relocation of racks. This relationship especially arises in large refrigerated warehouses (e.g., see SSI Schäfer, 2013), where the storage and retrieval vehicle moves fast whereas automatically relocating the massive racks takes up to several minutes. In this case, considerable savings of picking time can be realized compared to widespread rules-of-thumb often applied for picking order sequencing.
- The effect of providing additional open aisles quickly diminishes. Thus, it should be carefully evaluated whether the increasing picking performance can indeed justify the additional space occupied by additional open aisles. While two or three open aisles may be suitable layout alternatives, too many of them run counter to the intention of providing high-density storage with mobile racks.

Future research should tackle the case where multiple open aisles are simultaneously available. Especially, exact solution procedures constitute a challenging task. Furthermore, the two-sided access of racks via both adjacent aisles reported in Chang et al. (2007) and Hu et al. (2009) considerably impacts the problem structure and, therefore, justifies further research effort. Finally, also related operational problems, e.g., storage assignment, and layout problems have not been considered for mobile rack systems.

Appendix A. Computational complexity of OS-single

This appendix provides a detailed proof of NP-hardness in the strong sense of OS-single. We prove this complexity status by a reduction from the Hamiltonian Path problem (HPP), which is well-known to be strongly NP-complete (Garey & Johnson, 1979) and can be stated as follows. Given an undirected graph $G = (V, E)$ the HPP asks whether or not a path through the graph exists that visits all vertices $v \in V$ exactly once. Note that this problem remains NP-complete if a fixed start node for the path is given, see Garey and Johnson (1979).

Proof. Consider an instance I of HPP given by graph $G = (V, E)$ and start node p. Let d_p be the degree of node p. We construct an instance I' of OS-single in polynomial time as follows. We introduce $n = |V| - 1$ picking orders and $m = |E| - d_p + 1$ aisles as follows. We have a picking order for each node but p. Furthermore, we have a distinct aisle for each edge connecting nodes in $V \backslash \{p\}$. The aisle corresponding to $(v, v') \in E$ with $v \neq p \neq v'$ is to be visited only for processing the picking orders corresponding to v and v'. Finally, we have a single aisle a_p corresponding to those edges connecting p to the nodes in $V \backslash \{p\}$ which is the initially open aisle. This aisle is to be visited only for processing picking orders corresponding to nodes connected to p.

Picking orders to be processed in I' correspond to the nodes to be visited in I. If and only if there is an edge connecting nodes v and v' the picking orders corresponding to these nodes have an aisle to be visited in common. Thus, if two nodes can be visited consecutively in a Hamiltonian path of graph G the corresponding picking orders can be processed consecutively reusing one aisle.

We will now show that I is a yes-instance if and only if there is a feasible solution to I' implying $2(m-1) + d_p - n$ aisle relocations.

If I is a yes-instance, processing the picking orders in the sequence of the corresponding Hamiltonian path implies a solution to I' with $2(m-1) + d_p - n$ aisle relocations. For each edge $(v, v') \in E$, $v \neq p$, $v' \neq p$, used in the Hamiltonian path the corresponding pair of consecutive picking orders have an aisle to be visited in common. Furthermore, recall that this aisle is not to be visited for any other picking order. Hence, this aisle can be chosen to be the last one visited while processing the picking order corresponding to v and to be the first one visited while processing the picking order corresponding to v'. Similarly, $(p, v) \in E$ used in the Hamiltonian path implies that processing the picking order corresponding to v requires visiting the initially open aisle. This aisle then can be chosen to be the first one while processing the picking order corresponding to v. This gives us $2(|E| - d_p) - (|V| - 2)$ aisle relocations for aisles different from the initially open aisle. For the initially open aisle we have $d_p - 1$ relocations. In total, thus, we have

$$2(|E| - d_p) - (|V| - 2) + d_p - 1$$
$$= 2(|E| - d_p) + d_p - (|V| - 1) = 2(m-1) + d_p - n$$

aisle relocations.

If there is a feasible solution π to I' implying $2(m-1) + d_p - n$ aisle relocations, then the first aisle visited for the first picking order in the sequence must be the initially open aisle. Furthermore, for each pair of consecutive picking orders the last aisle visited for the first picking order must coincide with the first aisle visited for the second picking order. Hence, the nodes corresponding to consecutive picking orders must be connected by an edge and the node corresponding to the first picking order must be connected to p. □

Example 3. Fig. 11 exemplifies the reduction used in the proof. Graph G is given as $V = \{1, 2, 3, 4, 5\}$ and $E = \{\{1, 4\}, \{1, 5\}, \{2, 3\}, \{2, 5\}, \{3, 4\}, \{3, 5\}\}$. Instance I is specified by G and $p = 5$. Instance I' consequently has $n = 4$ picking orders and $m = 4$ aisles, namely $1, \dots, 4$. Furthermore, we have $A_1 = \{1, 4\}$, $A_2 = \{2, 4\}$, $A_3 = \{2, 3, 4\}$, and $A_4 = \{1, 3\}$. A sequence of picking orders corresponding to a Hamiltonian path is (1, 4, 3, 2) and implies $2(m-1) + d_p - n = 5$ aisle relocations.

Appendix B. Larger example for DP with given order sequence

This appendix gives an example for the more efficient version of DP (see Section 2.2.2), which determines the first and last open aisles for a given picking order sequence.

We consider a mobile rack warehouse with $m = 9$ aisles and $n = 8$ picking orders to be processed. The aisles to be visited per picking order are defined as follows: $A_1 = \{1, 2\}$, $A_2 = \{3, \dots, 6\}$, $A_3 = \{4, \dots, 8\}$, $A_4 = \{1, \dots, 6\}$, $A_5 = \{5, \dots, 8\}$, $A_6 = \{3, \dots, 6\}$, $A_7 = \{1, \dots, 3\}$, and $A_8 = \{4\}$. Initially, aisle $\delta = 9$ is open and the predetermined sequence of picking orders is $\sigma = \langle 1, 2, 3, 4, 5, 6, 7, 8]$. Fig. 12 depicts the DP graph as proposed by the first DP applying states (i, a). In order to keep the picture simple we abstain from drawing each single transition. The structure can be summarized as follows. Each state, which is a starting point for a transition of cost zero, has exactly $r - 1$ such transitions (with cost zero) where r is total number of states in the next stage. The only other transition, i.e., one with cost 1, leaving such a state (i, a) reaches state $(i + 1, a)$. States that are starting point for a transition of cost zero in this example are (2, 4), (2, 5), (2, 6), (3, 4), (3, 5), (3, 6), (4, 5), (4, 6), (5, 5), (5, 6), and (6, 3). All remaining states have only outgoing transitions of cost 1.

Now, Fig. 13 depicts the alternative DP graph as proposed for the second DP. The previous graph is shrunk to the second one as follows. For each stage i we have a state $(i, \bar{S}(i))$ representing those original states having only outgoing transitions of cost 1. We

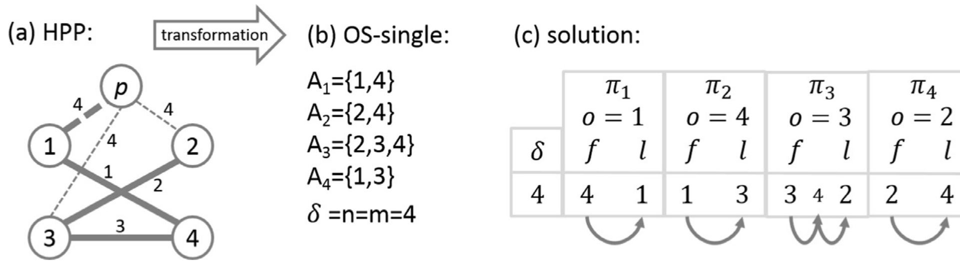


Fig. 11. Transformation for Example 3 and corresponding feasible solutions.

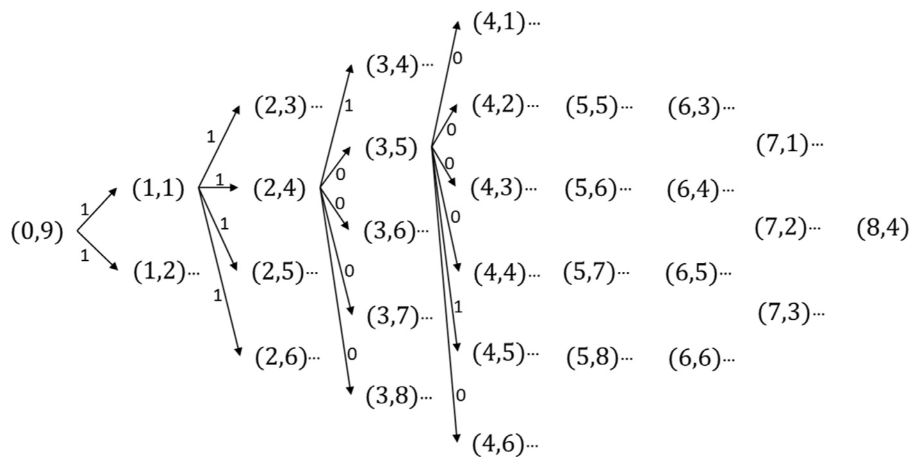


Fig. 12. Large graph for first DP.

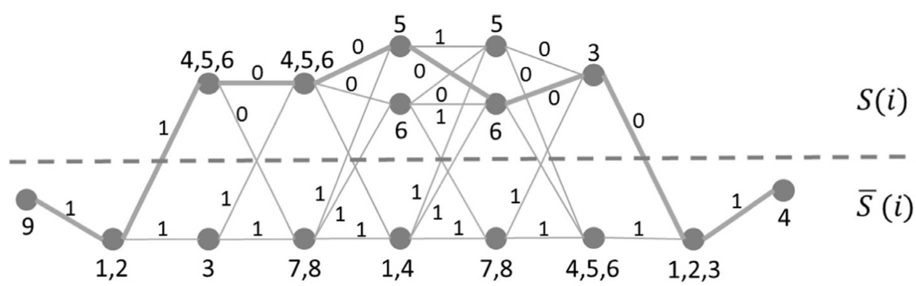


Fig. 13. Condensed graph for second DP.

depict these states at the very bottom of the graph and list the set of last aisles corresponding to them next to the corresponding node. In the upper part of the graph we have nodes corresponding to original states, which have an outgoing transition of cost 0. In stages 2 and 3, we have states (2, S(2)) and (3, S(3)) since we have at least three states with outgoing transitions of cost 0 in these stages of the original DP. We, again, list the set of last aisles corresponding to these states next to their aggregated nodes. In stage 4 (5), we have states $(4, a_1^4)$ and $(4, a_2^4)$ ($(5, a_1^5)$ and $(5, a_2^5)$) representing the two corresponding original states. Finally, in stage 6, we have state $(6, a_1^6)$ representing the only such original state.

The bold-faced path within Fig. 13 represents an optimal solution. Once we have determined such a minimum cost path we need to extract the optimum path for the original DP.

1. Reaching state $(1, \bar{S}(1))$ of the condensed DP is interpreted as reaching either state (1, 1) or state (1, 2) of the first DP. A random choice can decide between both states, because they neither differ in the cost for reaching them nor in the cost for leaving them.
2. Reaching (2, S(2)) is interpreted as reaching (2, 4), (2, 5), or (2, 6). Again, we can choose one of them randomly, since we reach (3, S(3)) next, which means that we can reach at least two original states represented (3, S(3)) using transitions of cost 0, no matter which original state represented by (2, S(2)) is chosen.
3. Reaching (3, S(3)) is interpreted as reaching (3, 4), (3, 5), or (3, 6). Note that we cannot chose (3, 5) since we next definitely reach original state (4, 5) using a transition of cost 0. Consequently, we choose (3, 4) or (3, 6).
4. Reaching $(4, a_1^4)$, $(5, a_2^5)$, and $(6, a_1^6)$ is interpreted as reaching (4, 5), (5, 6), and (6, 3).
5. Reaching $(7, \bar{S}(7))$ is interpreted as reaching (7, 1) or (7, 2) (chosen randomly). Note that we cannot chose (7, 3) since we cannot reach it from (6, 3) using a transition of cost 0.
6. Reaching the final state is interpreted as reaching (8, 4).

References

Allahverdi, A., Ng, C. T., Cheng, T. E., & Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research, 187*, 985–1032.

Bartholdi, J. J., III, & Hackman, S. T. (2014). *Warehouse and distribution science*. Supply Chain and Logistics Institute. Release 0.96

Błażewicz, J., Ecker, K., Pesch, E., Schmidt, G., & Węglarz, J. (2007). *Handbook on scheduling*. Berlin: Springer.

Boysen, N., Boywitz, D., & Weidinger, F. (2016). *Deep-lane storage of time-critical items: One-sided vs. two-sided access*. Friedrich-Schiller-University Jena. Working Paper.

Boysen, N., & Fliedner, M. (2010). Determining crane areas in intermodal transshipment yards: The yard partition problem. *European Journal of Operational Research, 204*, 336–342.

Chang, T. H., Fu, H. P., & Hu, K. Y. (2007). A two-sided picking model of m-AS/RS with an aisle-assignment algorithm. *International Journal of Production Research, 45*, 3971–3990.

Chisman, J. A. (1975). The clustered traveling salesman problem. *Computers and Operations Research, 2*, 115–119.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability*. New York: Freeman.

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

Gu, J., Goetschalckx, M., & McGinnis, L. F. (2010). Research on warehouse design and performance evaluation: A comprehensive review. *European Journal of Operational Research, 203*, 539–549.

Gue, K. R., & Kim, B. S. (2007). Puzzle-based storage systems. *Naval Research Logistics, 54*, 556–567.

Held, M., & Karp, R. M. (1962). A dynamic programming approach to sequencing problems. *SIAM Journal, 10*, 196–210.

Hu, K. Y., Chang, T. H., Fu, H. P., & Yeh, H. (2009). Improvement order picking in mobile storage systems with a middle cross aisle. *International Journal of Production Research, 47*, 1089–1104.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*, 671–680.

de Koster, R., Le-Duc, T., & Roodbergen, K. J. (2007). Design and control of warehouse order picking: A literature review. *European Journal of Operational Research, 182*, 481–501.

Osman, I. H., & Kelly, J. P. (2012). *Meta-heuristics: Theory and applications*. New York: Springer.

Pinedo, M. (2012). *Scheduling: Theory, algorithms, and systems*. New York: Springer.

Russon, D., Kútik, V., & Clarke, A. (1982). Future storage requirements at the British library lending division: Fixed or mobile shelving? *Interlending and Document Supply, 10*, 54–58.

SSI Schäfer (2013). Vollautomatisierte Verschieberegalanlage (in German). http://media.ssi-schaefer.de/fileadmin/ssi/documents/navigationsbaum/logistiksysteme/vollautomatische_systeme/verschieberegallager/vollautomatisierte_v_de.pdf (last access: September 2016).

Yu, Y., & de Koster, R. B. (2009). Optimal zone boundaries for two-class-based compact three-dimensional automated storage and retrieval systems. *IIE Transactions, 41*, 194–208.

Zaerpour, N., Yu, Y., & de Koster, M. B. M. (2016). Small is beautiful: A framework for evaluating and optimizing live-cube compact storage systems. *Transportation Science*. in press.

Zaerpour, N., Yu, Y., & Koster, R. (2015). Storing fresh produce for fast retrieval in an automated compact cross-dock system. *Production and Operations Management, 24*, 1266–1284.

Zhao, J., Choa, V., & Broms, B. B. (1996). Construction and utilization of rock caverns in Singapore part b: Development costs and utilization. *Tunnelling and Underground Space Technology, 11*, 73–79.