A Linear-Size Model for the Single Picker Routing Problem with Scattered Storage

Laura Lüke[a], André Hessenius[a,*], Stefan Irnich[a]

[a] *Chair of Logistics Management, Department of Business & Economics, Johannes Gutenberg University Mainz, Jakob-Welder-Weg 9, D-55128 Mainz, Germany.*

Abstract

We present a new approach of solving the single picker routing problem with scattered storage (SPRP-SS), which is a fundamental problem in modern warehouse operations management. The SPRP-SS assumes that SKUs of articles are stored at possibly many locations. An effective integer programming based approach relies on extending the state space of Ratliff and Rosenthal's dynamic program for the basic single picker routing problem to accommodate the SPRP-SS. As a result, the mixed integer linear programming (MIP) formulation has a quadratic number of variables. We propose two modifications of the extended state space to retain the linearity of the models. This is achieved by replacing the quadratically growing parallel edges of the extended state space by two different types of linear-size subnetworks. These replacements lead to different state spaces and herewith different MIP formulations, for which we analyze theoretical properties such as their size and strength of the linear relaxations. We compare the new formulations with the state of the art using a collection of 800 SPRP-SS instances. The results show that the new formulations are more than competitive providing integer optimal solutions of realistic and even large-scale instances in less than two seconds on average. The second formulation outperforms the current one regarding the computational speed: For the largest instances with 200 articles to be collected, average speedups reach the factors of 3.18 and 4.87 for general and unit demand, respectively.

Keywords: routing; warehousing; picker routing; scattered storage

1. Introduction

We consider order picking operations in warehouses where pickers travel through the warehouse to collect demanded articles from the storage locations (picker-to-parts). Order picking accounts for more than half of the warehouse operating costs (Bartholdi and Hackman, 2019) and therefore represents a good opportunity for expense optimization. We address the underlying single tour routing problem, which is known as the *single picker routing problem* (SPRP) or *order picking problem.* The most basic version of the SPRP considers a rectangular single-block parallel aisle warehouse with a single depot, which is start point and end point of each tour. The seminal work of Ratliff and Rosenthal (1983) shows that a minimum-length picker tour can be computed with dynamic programming in linear time. More precisely, the overall computational effort is linear in the number of aisles and the number of storage locations (=pick positions) to visit, as shown by Heßler and Irnich (2022).

Several variants of the SPRP extend the basic problem and require the computation of picker tours for different warehouse layouts (two-block (Roodbergen and de Koster, 2001), multi-block (Pansart et al., 2018), fishbone (Çelk and Süral, 2014), butterfly (Öztürkoğlu et al., 2012), etc.) and characteristics of the start and drop-off point(s) of a picker tour (Masae et al., 2020). In addition, simpler variants of the SPRP

[*] Corresponding author.

Email addresses: lueke@uni-mainz.de (Laura Lüke), hessenius@uni-mainz.de (André Hessenius), irnich@uni-mainz.de (Stefan Irnich) result from routing policies such as traversal (a.k.a. S-shape), midpoint, largest gap (Hall, 1993), return, and composite (Petersen, 1997). Routing policies address the practical issue that human pickers may not be able to execute all types of optimal tours, which can be complicated, counterintuitive, and difficult to memorize. As a result, the picker must execute a tour defined by a few simple rules. An more detailed overview of SPRP variants and pointers to the literature can be found in (Heßler and Irnich, 2024, Table 1).

In this paper, we address the *SPRP with scattered storage* (SPRP-SS), which is another variant of SPRP. The term scattered storage describes the storage strategy of placing the *stock keeping units* (SKUs) of one or more articles not only at a single, but possibly several pick positions in the warehouse (this is also our definition: a SKU is one item/a single unit of an article, where identical SKUs can be stored at different locations/positions). The reasoning behind scattering is that SKUs of a demanded article can be picked up quickly no matter of the picker's current position in the warehouse (Weidinger, 2018). Scattered storage is the predominant storage strategy in modern e-commerce warehouses of companies like Amazon or Zalando (Weidinger, 2018; Weidinger et al., 2019; Boysen et al., 2019; Khan et al., 2024).

In contrast to the long history of the basic SPRP, the SPRP-SS has only received increased attention since the middle of the last decade. First addressed by Singh and van Oudheusden (1997) and Daniels et al. (1998), up to this day, all competitive exact solution approaches are *mixed-integer programming* (MIP) based except for the Benders approach of Haouassi et al. (2024). Recently, Goeke and Schneider (2021) proposed an effective MIP formulation that solves instances of the SPRP-SS of realistic size to proven optimality. This approach is only outperformed by another MIP-based solution approach of Heßler and Irnich (2024). Their idea is as follows: In the state space of the dynamic-programming (DP) approach of Ratliff and Rosenthal (1983) for the basic SPRP, every feasible picker tour is a path and vice versa. The first property also holds for the SPRP-SS. With an extension in the underlying state space, the authors use a shortest path formulation to solve the problem. Herein, the extension of the state space requires the addition of new edges. The number of edges grows quadratically with the number of relevant pick positions per aisle.

There exist some other approaches for the SPRP-SS: Weidinger (2018) compared a decomposition procedure (select the pick positions by different priority rules and use the algorithm of Ratliff and Rosenthal to determine a picker tour) with the MIP of Daniels et al. (1998) complemented with MTZ-based subtour-elimination constraints (Miller et al., 1960). A rather involved MIP-based approach has been presented by Su et al. (2023) for multi-block parallel-aisle warehouses. Unfortunately, this approach has not been compared with any other approach, e.g., using the fact that the SPRP-SS with unit demand can be modeled and solved as a *generalized TSP* (GTSP) for any warehouse layout (for the definition of *unit demand* and *general demand*, see Section 2). Wildt et al. (2024) exactly use this fact, which makes their approach generally applicable, but ignores that warehouses often have a well-structured layout that can be exploited algorithmically. Accordingly, they use a series of transformations (SPRP-SS to GTSP, GTSP to clustered TSP, clustered TSP to asymmetric TSP, asymmetric to symmetric TSP) to finally solve them as ordinary TSP instances, either with a branch-and-cut-based TSP solver like Concorde (Applegate et al., 2003) or with the Lin–Kernighan–Helsgaun (LKH) heuristic (Helsgaun, 2000). The equivalence to the GTSP was also discussed by Heßler and Irnich (2024) who compare a re-implementation of the branch-and-cut algorithm of Fischetti et al. (2002) with their approach showing that for small pick lists the GTSP-based solution can be competitive. Note that for general demand the equivalence to the GTSP is no longer given, so that Wildt et al. (2024) proposed heuristic transformations for this case.

1.1. Contributions

We introduce two new MIP formulations for the SPRP-SS which are linear in the number of aisles and in the number of pick positions. We describe both formulations now:

EG: The first formulation is formally identical to the network-flow model of Heßler and Irnich (2024), but uses a new underlying state space which replaces parallel edges (a quadratic number of gap actions is possible) by a linear-size subnetwork. Since the subnetwork allows to expand gaps, this formulation is called *enlarged gaps* (EG) formulation.

RG: The second formulation builds a different kind of linear-size subnetwork in which gaps can only be reduced. Accordingly, the second formulation is called the *reduced gaps* (RG) formulation. In case of general demand, the network-flow model of Heßler and Irnich (2024) must be modified resulting in a slightly larger model with auxiliary variables and additional constraints. However, these model extensions keep the formulation linear.

We compare the two formulations among each other and against the state-of-the-art formulation of Heßler and Irnich (2024), in the following referred to as formulation HI. The theoretical and computational analyses provide the following findings:

- Both formulations EG and RG have a generally weaker linear relaxation than the formulation HI. On the large SPRP-SS instance set used in the computational studies, formulation RG is almost as strong as formulation HI, while formulation EG is weaker.
- Both formulations EG and RG give significantly smaller MIP models than formulation HI. Formulation RG reduces, on average, more variables/edges (between 48 and 86 percent) than formulation EG (between 34 and 81 percent) in comparison to the quadratically sized networks used in formulation HI.
- Compared to the state of the art, formulation HI, the new formulation RG accelerates the MIP solution by the factor of 1.89 for general demand and by the factor of 2.42 for unit demand for the SPRP-SS instances in the testbed.

1.2. Structure

The remainder of this paper is structured as follows: In Section 2, we formally introduce the SPRP-SS. We summarize the relevant solution approaches in Section 3 which includes the DP of Ratliff and Rosenthal for the SPRP, the extension of the state space of Ratliff and Rosenthal to incorporate aisle traversal options needed for the SPRP-SS, and the network-flow model of Heßler and Irnich, which is finally used to solve SPRP-SS instances. The subnetworks for formulations EG and RG are presented in Section 4. Computational analyses and their results are discussed in Section 5. We close the paper by drawing final conclusions in Section 6.

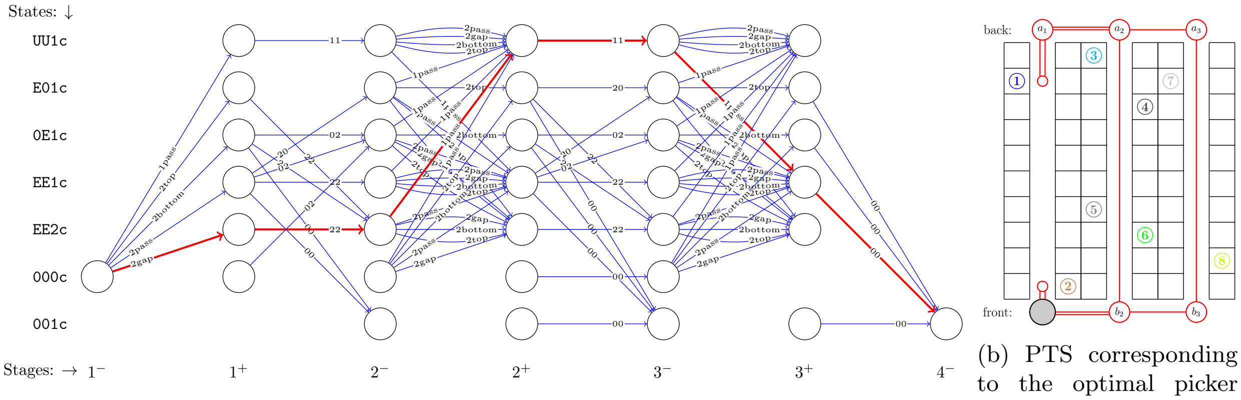
2. The Single Picker Routing Problem with Scattered Storage

In the SPRP-SS, the set S denotes the different *articles*, where each article $s \in S$ is stored at (possibly several) pick positions $p \in P_s$. The set $P = \bigcup_{s \in S} P_s$ describes all relevant pick positions. At position $p \in P_s$, $b_{sp} \geq 1$ units of article $s \in S$ are available for being collected. The SKUs are stored at the pick positions along both sides of the picking aisles in a single-block parallel-aisle warehouse. Note that the terms SKU and article describe the same object with article being what is requested by a customer and SKU referring to a unit of an article stored in the warehouse. The goal of the SPRP-SS is then to find a minimum-length picker tour that starts and ends at the given I/O point 0 and visits a subset of the pick positions such that the given demand q_s for each article $s \in S$ is satisfied, i.e., can be collected from the positions visited. Daniels et al. (1998) already stated that apart from the *general-demand case* with $q_s > 1$ for at least some articles $s \in S$, the SPRP-SS can also emerge in the so-called *unit-demand* case. In this case, the demanded number is $q_s = 1$ for all articles $s \in S$. The SPRP-SS with unit demand can be modeled as a *generalized traveling salesman problem* (GTSP, Fischetti et al., 2002) with the pairs in $\{(s,p) \in S \times P : b_{sp} = 1\}$ as the cities and the sets $C_s = \{(s,p) : p \in P_s\}$ for $s \in S$ as the clusters (Daniels et al., 1998). The unit-demand case is equivalent to the situation where each pick position b_{sp} holds sufficient supply to cover the whole demand q_s of the respective article $s \in S$.

The SPRP-SS is an NP-hard problem (Weidinger, 2018). This statement remains true when picker tours are restricted to routing policies such as traversal, return, midpoint, largest gap, and composite (Lüke et al., 2024). Approaches for the exact solution and optimal routing have been proposed by Weidinger (2018), Goeke and Schneider (2021), and Heßler and Irnich (2024). The latter approach is the best-performing exact solution algorithm to date. Since our goal is to improve on the model proposed by Heßler and Irnich, we will present their formulation and approach in the following.

3. Exact Solution of the SPRP and SPRP-SS

We start with the description of the DP by Ratliff and Rosenthal (1983), explain how the state space of the DP was extended by Heßler and Irnich (2024) to include the option of not necessarily having to visit all



(a) State space (V, E)

(b) PTS corresponding to the optimal picker tour

Figure 1: State space and PTS for a warehouse with aisles $J = \{1, 2, 3\}$ and articles $S = \{1, 2, \dots, 8\}$ (unit demand). The highlighted path (red/bold) in the state space represents the optimal picker tour.

pick positions, and present their binary formulation of the SPRP-SS.

3.1. Basic State Space of Ratliff and Rosenthal

We briefly introduce the ideas and assumptions of Ratliff and Rosenthal (1983) that allow to formulate the SPRP and solve it by DP in linear time. They consider a single-block parallel-aisle warehouse. Each aisle is uniformly divided into a number of pick positions of identical size. Each demanded article is stored at a pick position in the warehouse resulting in a set of pick positions P that the picker necessarily has to visit. Picking from either the right-hand side or the left-hand side (or both) is not distinguished, so that the side of the aisle is irrelevant for computing the picker tour length. Therefore, different SKUs located at the same pick position can be seen as a single picking request.

Formally, let $J = \{1, 2, \dots, m\}$ denote the aisles in a single-block rectangular warehouse with parallel aisles. A picker tour can be constructed by alternately deciding how the picker traverses an aisle and how he or she processes from aisle j to aisle $j+1$. Moving through an aisle $j \in J$ results in the transition from stage j^- to j^+ while performing a *cross-aisle action* from an aisle j to $j+1$ leads to the transition from stage j^+ to $(j+1)^-$. The picker tour ends in a finite state at stage $(m+1)^-$, so that the state space of the DP has $2m+1$ stages. To model the states and transitions of the DP, Ratliff and Rosenthal introduce so-called *partial tour subgraphs* (PTSs). For an aisle $j \in J$, a PTS represents the parts of the picker tour from aisle 1 to aisle j and specifies the vertex parity of the vertices a_j and b_j located at the back and front of each aisle $j \in J$, see Figure 1b. The vertex parities in combination with the number of connected components of the PTS characterize the states of the DP. Ratliff and Rosenthal show that only seven states are relevant for optimal picker tours. These states can be denoted as

$$\mathscr{S} = \{\texttt{UU1c}, \texttt{0E1c}, \texttt{E01c}, \texttt{EE1c}, \texttt{EE2c}, \texttt{000c}, \texttt{001c}\}.$$

Here, the first (second) symbol indicates the parity of the vertex a_j (b_j) with `0`, `U`, and `E` standing for not reached, odd (=uneven) degree, and even degree, respectively. The number of connected components is shown by the last two symbols: The state `0c` indicates an empty PTS, `1c` one with a single connected component, and `2c` a subgraph with two connected components. Figure 1 shows an optimal picker tour with its corresponding PTS and state space (V, E). Within an aisle, six different *aisle actions* can be performed, namely

$$E_j^{aisle} = \{\texttt{1pass}, \texttt{2pass}, \texttt{top}, \texttt{bottom}, \texttt{gap}, \texttt{void}\}.$$

Action `1pass` (`2pass`) describes a single (double) traversal through the aisle (either direction), `top` (`bottom`) stands for a traversal from the back (front) cross-aisle to the lowest (highest) relevant pick position and back, and `gap` for entering the aisle from both sides leaving a maximum length gap in the middle while visiting all pick positions. When having an empty aisle, `void` represents no traversal through the aisle.

Regarding the *cross-aisle action* from aisle j^+ to $(j+1)^-$, there are five different options to consider:

$$E_j^{cross} = \{\texttt{00}, \texttt{11}, \texttt{20}, \texttt{02}, \texttt{22}\}.$$

The first (second) digit gives the number traversals of the back (front) cross-aisle. Note that every o-d-path in the state graph represents a feasible solution for the SPRP, where the origin and destination are the states $o = \texttt{000c}$ at stage 0^- and $d = \texttt{001c}$ at stage $(m+1)^- = 4^-$, respectively. Moreover, each edge $e \in E_j^{cross} \cup E_j^{aisle}$ is associated with a cost describing the length of the component added to the tour when performing the corresponding action. Heßler and Irnich (2022) have shown that the calculation of these costs can be done with an effort of $\mathcal{O}(m+n)$ with $n = |P|$ while solving the DP also takes linear effort with the same complexity (Ratliff and Rosenthal, 1983).

3.2. Scattered Storage and the Extended State Space

For the SPRP-SS, some additional notation is required, since each article may now be stored at several locations instead of having one unique pick position. Recall that q_s denotes the quantity of article $s \in S$ to be collected and b_{sp} denotes the supply (=number of SKUs) of $s \in S$ at pick position $p \in P$. Furthermore, let P_e denote the set of positions visited by an aisle traversal $e \in E_j^{aisle}$ in aisle $j \in J$. Thus, $b_{se} = \sum_{p \in P_e} b_{sp}$ is the quantity of article $s \in S$ that can be collected when traversing the aisle via e. Edges e with a non-negative supply of article $s \in S$ are denoted by E_s. For the sake of clarity, we distinguish between three types of articles:

(O) articles available in several aisles in the warehouse,
(UA) articles $s \in S$ available in a *unique aisle*, but with $|P_s| > 1$, and
(UP) articles that are only available at one *unique position* in the warehouse.

Note that looking at an arbitrary position in an aisle, different SKUs can be placed on the right-hand side, left-hand side or, when stored in shelves, even vertically over one another. Still, different SKUs located at one position in any of the described manners refer to the same pick position $p \in P$. Therefore, the number

$$n = |\bigcup_{s \in S} P_s| = |\{p \in P : \exists s \in S \text{ with } b_{sp} > 0\}| \tag{1}$$

of relevant positions can be smaller than the number of pairs $(s,p) \in S \times P$ with a positive supply $b_{sp} > 0$.

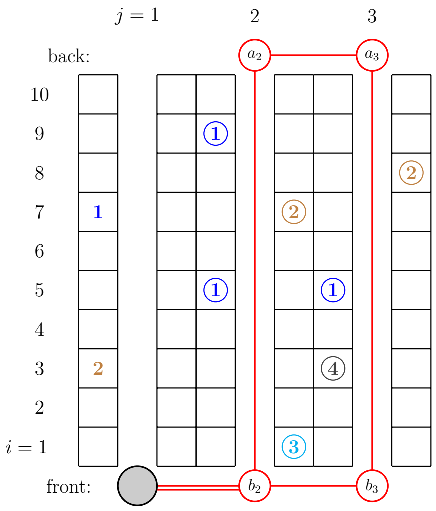
To incorporate the flexibility of not necessarily having to traverse all pick positions with demanded articles in the warehouse, an extension of the state space of Section 3.1 is needed. Specifically, additional *aisle actions* of type `top`, `bottom`, `void`, and `gap` must be considered, while all other *aisle actions* and *cross-aisle actions* remain identical. For scattered storage with general demand, any aisle traversal $e \in E_j^{aisle}$ leaving out positions $P_j^{'} \subset P$ in aisle $j \in J$ is feasible if and only if

$$\sum_{(s,p) \in P_s : p \notin P_j^{'}} b_{sp} \geq q_s \qquad \forall s \in S \tag{2}$$

holds.

Example 1. *Figure 2a shows an instance of the SPRP-SS with three aisles $J = \{1,2,3\}$ and four articles $S = \{1,2,3,4\}$. For simplicity, we assume unit supply, i.e., $b_{sp} = 1$ for all $s \in S, p \in P_s$. Moreover, we assume demands $q_1 = 3, q_2 = 2$, and $q_3 = q_4 = 1$. The resulting additional aisle actions are listed in Figure 2b.*

For aisle $j = 1$, three additional aisle actions *are to be considered. Standing at the back (front), it is now possible to only traverse the aisle up to position $i = 7$ ($i = 3$), or skip the aisle entirely, since (2) holds for an aisle action leaving out the position where article 2 (article 1) or both are stored. These additional options yield in the aisle actions* `top`(7), `bottom`(3), *and* `void`. *For the second aisle, article 3 must be picked up, since it is only available in one aisle so that no* `void` *action can be performed here, while also preventing any additional* `top(i)` *traversal. Note that the additional* `bottom` *and* `gap` *options always traverse at least*



(a) Warehouse and optimal picker tour; pick operations are encircled.

Aisle	type of	additional aisle actions
$j = 1$	`top`(i)	cell $i = 7$
	`bottom`(i)	cell $i = 3$
	`void`	
$j = 2$	`bottom`(i)	cell $i \in \{5, 7\}$
	`gap`(h, i)	cells $(h, i) \in \{(1, 7), (1, 9), (5, 9)\}$
$j = 3$	`bottom`(i)	cell $i \in \{3, 5\}$
	`gap`(h, i)	cells $(h, i) = (3, 8)$

(b) Additional aisle actions compared to non-scattered storage.

Figure 2: An instance of the SPRP-SS. The pick list contains four articles $S = \{1, 2, 3, 4\}$ with unit supply and demands $q_1 = 3, q_2 = 2$, and $q_3 = q_4 = 1$.

one pick position of article 1, since (2) would be violated otherwise. Therefore, `bottom`(1) is not allowed. For aisle $j = 3$, article 4 is unique so that `void`, `top`(5), and `top`(8) become infeasible. Furthermore, all traversals `gap`(i, k) can be disregarded if i and k are neighboring positions with a non-maximal gap.

The number of additional aisle actions in the extended state space is dominated by those of type `gap`(h, i). Indeed, if a majority of the pick positions are located within one or a few aisles, there can be $\mathcal{O}(n^2)$ aisle actions of type `gap`(h, i). Therefore, since the original state graph had $\mathcal{O}(m)$ edges, the total number of edges in the extended state space is bounded by $\mathcal{O}(m + n^2)$.

Due to the extension of the state space, an arbitrary o-d-path does not necessarily represent a feasible solution to the SPRP-SS. The following network-flow formulation is considering this fact.

3.3. Network-Flow Formulation for the SPRP-SS

Formulation HI uses the above extended state space (V, E) to model the SPRP-SS as an o-d-shortest-path problem with additional *demand-covering constraints* (DCCs). Variables $x_e \geq 0$ indicate the (unit) flow for each edge $E = \bigcup_{j \in J}(E_j^{aisle} \cup E_j^{cross})$. Moreover, let c_e be the cost of edge $e \in E$, i.e., the length of the picker tour associated with the action described by e. This leads to the following binary model:

$$z_{\text{SPRP-SS}} = \min \sum_{e \in E} c_e x_e \tag{3a}$$

$$\text{subject to} \quad \sum_{e \in \delta^+(\sigma)} x_e - \sum_{e \in \delta^-(\sigma)} x_e = \begin{cases} +1, & \text{if } \sigma = o \\ -1, & \text{if } \sigma = d \\ 0, & \text{otherwise} \end{cases} \qquad \forall \sigma \in V \tag{3b}$$

$$\sum_{e \in E_s} b_{se} x_e \geq q_s \qquad \forall s \in S \tag{3c}$$

$$x_e \in \{0, 1\} \qquad \forall e \in E \tag{3d}$$

The objective (3a) minimizes the length of the picker tour. Flow conservation is modeled by constraints (3b), where $\delta^+(\sigma)$ and $\delta^-(\sigma)$ denote the set of edges leaving and entering vertex $\sigma \in V$, respectively. The

DCCs (3c) ensure that enough SKUs are collected for each article $s \in S$. The domain of the flow variables is given by (3d).

Note that constraints (3c) can be strengthened by replacing b_{se} with $\min\{q_s, b_{se}\}$ for all $s \in S$ (we always do this). Moreover, for each article $s \in S$ of type UA, constraints (2) ensures that each aisle action of the respective aisle collects a sufficient quantity of s. Therefore, constraints (3c) are redundant for articles of types UP and UA. Note that every article of type UP is associated to a unique aisle.

The size of formulation HI, i.e., model (3), can be summarized as follows:

Property 1. *Formulation HI has $\mathcal{O}(m+n^2)$ variables, $\mathcal{O}(m+a)$ constraints, and $\mathcal{O}(an^2)$ non-zero coefficients (recall that n denotes the number of possible pick positions and $a = |S|$ denotes the number of different articles to be collected).*

Proof. The number of variables coincides with the number of edges in the extended state space which grows in the square of the relevant pick positions and is bounded by $\mathcal{O}(m+n^2)$. The number of constraints is bounded by $\mathcal{O}(m+a)$, where constraints (3b) contribute the summand m and constraints (3c) the summand a. The number of non-zero coefficients is bounded by $\mathcal{O}(m+an^2)$, since the incidence matrix has exactly $2|E| = \mathcal{O}(m+n^2)$ non-zeros, and the demand-covering constraints (3c) have $\sum_{s\in S} |E_s| = \mathcal{O}(an^2)$ non-zero coefficients. □

Additionally, Heßler and Irnich (2024) define dominance rules to reduce the number of parallel edges in E. Dominance leads to smaller state spaces and herewith a reduced number of variables in formulation HI. We also apply dominance to reduce the edges in the state space and to accelerate the solution of formulation HI by the MIP solver. Even with dominance, Property 1 is valid and it is straightforward to construct SPRP-SS instances for which the presented bounds are sharp.

4. Linear-Size State Spaces

As stated in Property 1, the number of edges in the state space of Heßler and Irnich is bounded by $\mathcal{O}(m+n^2)$. It grows quadratically with the number n of relevant pick positions, as defined in Eq. (1). Only the actions `gap` are responsible for this quadratic growth. Thus, we replace all parallel edges between states connected via actions `gap` by a small subnetwork of linear size. This can be done in different ways. We now present two types of subnetworks of linear size that can be included in the extended state space of formulation HI to fully replace all of the `gap` actions.

4.1. Enlarged Gaps Network

Let $\sigma, \sigma' \in V$ be two vertices connected with an action `gap` in the state space $G = (V, E)$. Moreover, let $j \in J$ be the associated aisle and let I_j denote the set of all relevant pick positions in this aisle (numbered from 1 to C, bottom up). Depending on the state of σ also the actions `top`, `bottom`, and `void` may connect σ with σ', i.e., they are parallel edges. Figure 1a shows the details:

- For $\sigma \in \{$`UU1c`, `EE1c`, `EE2c`$\}$, actions `gap`, `top`, and `bottom` connect to the same vertex σ'.
- For $\sigma =$ `0E1c`, only action `top` connects to the same vertex σ' as action `gap`.
- For $\sigma =$ `E01c`, only action `bottom` connects to the same vertex σ' as action `gap`.
- For $\sigma =$ `000c`, neither `top` nor `bottom` connects to the same vertex as `gap`.

The subnetwork that we construct includes all of the above parallel edges.

The basic idea is that two actions `top`(h) and `bottom`(i) are performed sequentially in the subnetwork so that they replicate the respective action `gap`(h, i). The construction is done in three steps (Figure 3 shows an example of a subnetwork):

1. Additional vertices v_{hi} are introduced for each action `gap`(h, i) where $h \in I_j$ and $i \in I_j$ with $h < i$ are consecutive pick positions in the considered aisle j.

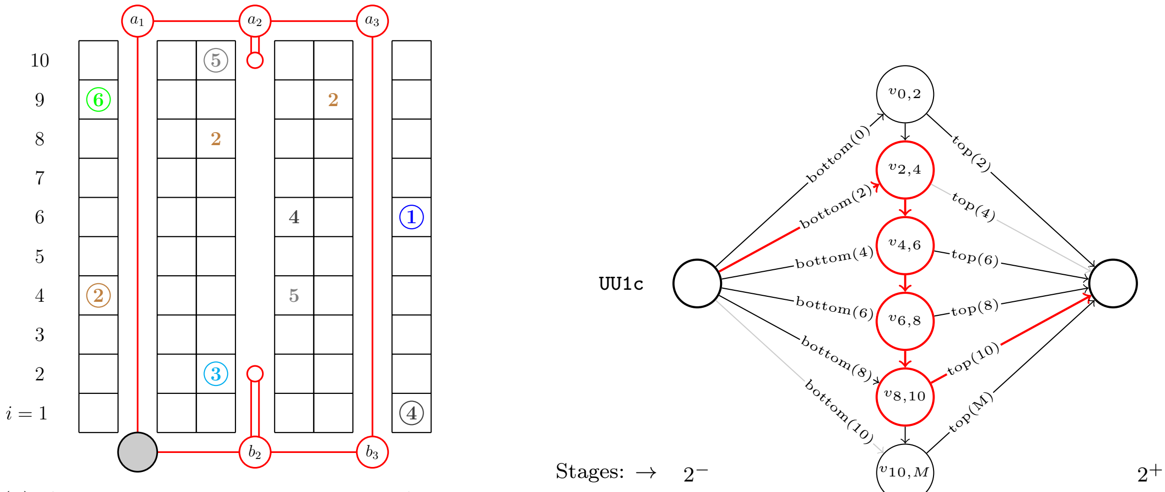
   Moreover, one or two vertices are added to replicate the original actions `bottom` and `top` connecting vertex σ with vertex σ' (see above for which type of state this is relevant). For the original action `bottom`, the vertex $v_{i_{\max},M}$ is added where $M > i_{\max} = \max I_j$. For the original action `top`, the vertex $v_{0,h_{\min}}$ is added where $h_{\min} = \min I_j > 0$. Note that for $\sigma \in \{\texttt{UU1c}, \texttt{EE1c}, \texttt{EE2c}\}$ both vertices are included in the network. In case that the action `void` connects σ with σ', it can be omitted in the subnetwork.

2. For the actions `bottom`(h), edges are added to connect σ with v_{hi} (ingoing edges of v_{hi}). Similarly, edges for the actions `top`(i) are added to connect v_{hi} with σ' (outgoing edges of v_{hi}).

   Actions `bottom`(0) and `top`(M) have a cost of 0, where M and 0 indicate that the aisle is not entered from the top or the bottom, respectively.

3. Downward edges of cost 0 between the additional vertices created in the first step enable actions `gap` with more space between the turning points. As a result, an action `bottom`(h) can be combined with an action `top`(i) also for non-consecutive pick positions h and i.

For a given (original) state space $G = (V, E)$, we denote by $G' = (V', E')$ the state space that replaces all actions `gap`, `top`, `bottom`, and `void` that are represented by parallel edges in G by the respective subnetwork. By construction, the new state space G' has $\mathcal{O}(m + n)$ vertices and $\mathcal{O}(m + n)$ edges.



(a) A warehouse with six articles. Articles 2 and 4 are available in different aisles (O), article 5 has a unique aisle (UA), and articles 1, 3 and 6 have a unique pick position (UP).

(b) Subnetwork EG for aisle $j = 2$ and the state of type `UU1c`. The highlighted path (red/bold) corresponds to `gap`(2,10) in Figure 3a. Gray edges can be omitted from the subnetwork by dominance considerations.

Figure 3: Example for a network which can enlarge the gaps.

Example 2. *Figure 3 visualizes a warehouse with aisles $J = \{1, 2, 3\}$ and six articles $S = \{1, 2, \ldots, 6\}$ that must be collected (unit demand, i.e., $q_1 = \cdots = q_6 = 1$) together with the optimal picker tour. We consider the second aisle $j = 2$ with five possible pick positions and the resulting subnetwork for the state $\sigma =$ `UU1c`.*

Dominance considerations can reduce the number of edges in the subnetwork. For example, the edge for the action `top`(4) can be omitted, since article 5 would be collected twice, which is not necessary for unit demand. The same is true for `bottom`(10).

Figure 3b and Example 2 are also helpful to highlight differences when either the original state space G or the new state space G' that includes the subnetworks are used within model (3). In formulation HI (using G), the DCCs (3c) are redundant for articles of type UP and UA, i.e., $s \in \{1, 3, 5, 6\}$ in the example. However, the structure of the subnetwork allows to use invalid combinations of actions. For example, if articles $s = 3$ and $s = 5$ have no associated constraint (3c), then the action `top`(M) can be combined with action `bottom`(0), which represents `void` and is not feasible due to (2). As a result, model (3) defined over the new state space G' is only valid for the SPRP-SS when all constraints of type (3c) are present for all $s \in S$. This version of model (3) defines formulation EG that we compare against formulation HI (and a third one defined in the next subsection). Even with more DCCs in formulation EG, the new formulation is linear as stated in the following property:

Property 2. *Formulation EG is of linear size, and it has $\mathcal{O}(m+n)$ variables and $\mathcal{O}(m+n)$ constraints.*

Proof. The number of vertices and edges per subnetwork is proportional to the number of pick positions in the respective aisle. Hence, the state space G' has not more than $\mathcal{O}(m+n)$ vertices and edges. Assuming that $a = |S| \leq n$, formulation EG has $\mathcal{O}(m+n)$ variables and constraints. □

The size of a model is only one factor that impacts the performance of a MIP solver that tries to solve the model. A second factor is the strength of the linear relaxation. Comparing formulations HI and EG, we can expect that formulation EG has a weaker linear relaxation, since the additional DCCs but not the linear-size state space G' ensure demand coverage for articles of type UA and UP (this allows LP solutions in which, e.g., an action `gap` is combined with action `void` both with value 0.5). Hence, there exists a clear tradeoff between model size and model strength. Only empirical tests can tell which of both formulations is solved faster, see Section 5.

Property 3. *The DCCs (3c) in Formulation EG are redundant for articles $s \in S$ of type UA and UP.*

4.2. Reduced Gaps Network

As the subnetwork EG, also the second subnetwork RG is used to replace parallel edges between two vertices σ and σ' describing actions `gap` (and `top`, `bottom`, and `void` depending on the state of σ). The additional vertices of the RG network are constructed using the notion of *non-extensible gaps* (NEGs). A feasible action `gap`(h, i) is an NEG, if no other action `gap`(h', i') with $h' \leq h$, $i' \geq i$, and $(h', i') \neq (h, i)$ is feasible. This definition implies that if `gap`(h, i) and `gap`(k, ℓ) are NEGs, then either $h < k$ and $i < \ell$ or $h > k$ and $i > \ell$. Note that, depending on the state of σ, we also include the artificial positions 0 and M. Formally, this requires the consideration of state-dependent position sets I_j for each aisle $j \in J$. For state `000c`, we consider positions I_j without 0 and M; for state `E01c`, positions $I_j \cup \{0\}$; for state `0E1c`, positions $I_j \cup \{M\}$; for the remaining states `UU1c`, `EE1c`, and `EE2c`, positions $I_j \cup \{0, M\}$. For the sake of simplicity, we will refrain from further formalizing this detail.

Also the construction of the subnetwork RG is done in three steps (Figure 4 shows an example of a subnetwork):

1. Additional vertices w_{hi} are introduced for each NEG `gap`(h, i). Note that NEGs may have $h = 0$ or $i = M$.

2. For each $h \in I_j$ (or $h \in I_j \cup \{0\}$ for some states, see above), the edge for the action `bottom`(h) is added to connect σ with $w_{h'i'}$ (ingoing edges of $w_{h'i'}$) where (h', i') is the NEG with maximum index $h' \leq h$. Similarly, for each $i \in I_j$ (or $i \in I_j \cup \{M\}$ for some states, see above), the edge for the action `top`(i) is added to connect $w_{h'i'}$ with σ' (outgoing edges of $w_{h'i'}$) where (h', i') is the NEG with minimum index $i' \geq i$.

   Actions `top`(M) and `bottom`(0) have a cost of 0.

3. Upward edges of cost 0 between the vertices created in the first step enable actions `gap` with less space between the turning points. As a result, an action `bottom`(h) can be combined with an action `top`(i) also for pick positions h and i that do not correspond to an NEG.

For a given (original) state space $G = (V, E)$, we denote by $G^* = (V^*, E^*)$ the state space that replaces all actions `gap`, `top`, `bottom`, and `void` that are represented by parallel edges in G by the respective subnetwork. By construction, the new state space G^* has $\mathcal{O}(m+n)$ vertices and $\mathcal{O}(m+n)$ edges.

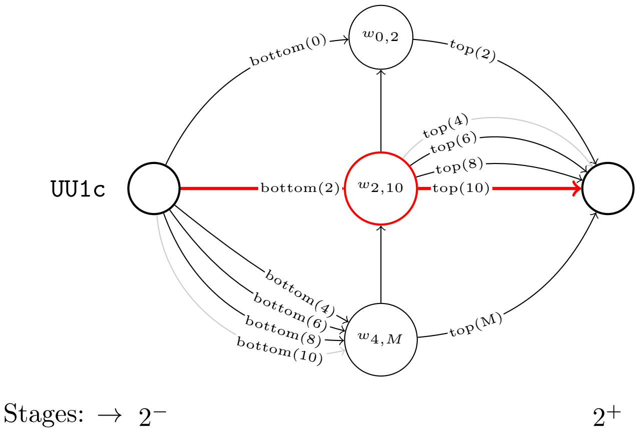


Figure 4: Subnetwork RG for aisle $j = 2$ and the state of type `UU1c`. The highlighted path (red/bold) corresponds to `gap`(2,10) in Figure 3a. Gray edges can be omitted from the subnetwork by dominance considerations.

Example 3. *(cont'd from Example 2) In Example 2, we focused on the aisle $j = 2$ and the state $\sigma =$ `UU1c` (see Figure 3). Under the unit-demand assumption, there are three NEGs (h, i), namely*

- $(0, 2)$ *corresponding to the action* `top`(2),
- $(2, 10)$ *corresponding to the action* `gap`$(2, 10)$, *and*
- $(4, M)$ *corresponding to the action* `bottom`(4).

The three NEGs are to be considered in the construction of the RG network, which is shown in Figure 4. We have three additional vertices w_{02}, $w_{2,10}$, and $w_{4,M}$. In addition, there are six edges for possible actions `bottom`(h) *and also six edges for possible actions* `top`(i)*. In both cases, four of the six possible actions are parallel edges in the subnetwork G^*.*

As in Example 2, dominance allows to eliminate actions `bottom`(10) *and* `top`(4).

Using model (3) together with the new state space G^* gives a formulation that is correct for articles s of type UA and UP. The feasibility of the NEGs with respect to conditions (2) (by definition of an NEG) ensures that every path through the subnetwork RG allows to collect a sufficient amount of these articles s. There is no need to have the DCCs (3c) for them, i.e., they are surely redundant for articles of type UA and UP. Hence, the new RG network heals one of the shortcomings that we identified for the EG network, see Section 4.1.

However, some possible paths in the subnetwork RG do not describe reasonable actions `gap`(h, i). In Figure 4, e.g., one of the actions `bottom`(h) for a cell $h \in \{4, 6, 8, 10\}$ could be traversed first, and the edge for `top`(2) for cell $i = 2$ afterwards. These actions "overlap" in the sense that the pick positions 2 and 4 (and depending on h some more cells) are reached from the top as well as from the bottom. When assigning a supply to these edges (this is what we do in model (3) when using the coefficients b_{se} in the DCCs (3c)), this supply would be counted twice when using such overlapping actions of the subnetwork RG. This is uncritical for unit demand, but leads to an incorrect formulation for general demand.

We can summarize these observations for using model (3) together with the new state space G^* defined over the RG network, denoted as Formulation RG$^{(3)}$.

Property 4. *Formulation RG$^{(3)}$ is a valid formulation for the SPRP-SS with unit demand. It is of linear size, and it has $\mathcal{O}(m+n)$ variables and $\mathcal{O}(m+n)$ constraints. The DCCs (3c) are redundant for articles $s \in S$ of type UA and UP.*

For the general-demand case, we present a modified MIP model for the SPRP-SS that uses two types of variables: As before, binary variables x_e for all edges in the state space describe the o-d-path. In addition, continuous variables y_{sp} describe, for each article $s \in S$ and possible pick position $p \in P_s$, the amount that is collected. By bounding $y_{sp} \leq b_{sp}$, it is ensured that supply from this position is not double counted. In addition, let E_p denote the set of all edges and corresponding actions that traverse position $p \in P$. The new model reads as follows:

$$
\begin{aligned}
& z_{\text{SPRP-SS}} = \min \sum_{e \in E} c_e x_e && && \text{(4a)} \\
\text{subject to} \quad & \sum_{e \in \delta^+(\sigma)} x_e - \sum_{e \in \delta^-(\sigma)} x_e = \begin{cases} +1, & \text{if } \sigma = o \\ -1, & \text{if } \sigma = d \\ 0, & \text{otherwise} \end{cases} && \forall \sigma \in V && \text{(4b)} \\
& \sum_{e \in E_p} b_{sp} x_e = y_{sp} && \forall s \in S,\, p \in P_s && \text{(4c)} \\
& \sum_{p \in P_s} y_{sp} \geq q_s && \forall s \in S && \text{(4d)} \\
& x_e \in \{0, 1\} && \forall e \in E && \text{(4e)} \\
& 0 \leq y_{sp} \leq b_{sp} && \forall s \in S,\, p \in P_s && \text{(4f)}
\end{aligned}
$$

The objective function (4a) and the constraints (4b) and (4e) remain identical to those of the first model (3). The DCCs (3c) are reformulated into (4c) and (4d) making sure that a sufficient amount is collected and no supply is double counted. The feasible domain for the supply variables is stated in (4f).

We denote by Formulation RG$^{(4)}$ the use of the new model (4) defined over the new RG network G^*. We summarize what we know about this formulation:

Property 5. *Formulation RG$^{(4)}$ is a valid formulation for the SPRP-SS (for both unit demand and general demand). It is of linear size, and it has $\mathcal{O}(m+n)$ variables and $\mathcal{O}(m+n)$ constraints. The DCCs are redundant for articles $s \in S$ of type UA and UP, i.e., the corresponding variables y_{sp} and constraints (4c), (4d), and (4f) can be omitted.*

Strength of Linear Relaxations. For comparing different formulations to the same problem, not only the size of the respective models is important. For unit-demand instances of the SPRP-SS, we will now show that the linear relaxation of formulation RG$^{(3)}$ is at least as strong as the one of formulation RG$^{(4)}$. Let $z_{\text{LP}}^{\text{RG}(3)}$ and $z_{\text{LP}}^{\text{RG}(4)}$ denote the objective value of the linear relaxation of formulation RG$^{(3)}$ and RG$^{(4)}$, respectively.

Property 6. *For all instances of the SPRP-SS with unit-demand, the relation $z_{LP}^{RG(3)} \geq z_{LP}^{RG(4)}$ holds.*

Proof. Recall that in the unit-demand case, $q_s = 1$ for all $s \in S$ as well as $b_{se} = 1$ and $b_{sp} = 1$ for all $e \in E_s$ and $p \in P_s$, respectively. Therefore, we can always use b_{se} instead of b_{sp} for the positions $p \in P_s$ an edge $e \in E_s$ visits.

Let the flow $\bar{\mathbf{x}} = (\bar{x}_e) \in [0,1]^E$ denote a feasible solution to the linear relaxation of RG$^{(3)}$. We can prove the statement by showing that this solution can be translated into a feasible solution to the linear relaxation of RG$^{(4)}$ with the same (or lower) cost. To this end, let

$$\bar{y}_{sp} = \min \left\{ 1, \sum_{e \in E_p} \bar{x}_e \right\}$$

for all $s \in S$ and $p \in P_s$.

With these definitions it is guaranteed that the solution $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ satisfies all constraints of (4), where only the fulfillment of constraints (4d) needs to be shown. We distinguish two cases for each article $s \in S$: If $\bar{y}_{sp} = 1$ for at least one of the positions $p \in P_s$, we trivially have $\sum_{p \in P_s} \bar{y}_{sp} \geq 1$, i.e., constraint (4d) holds for article s. Otherwise, $\min\{1, \sum_{e\in E_p} \bar{x}_e\} = \sum_{e\in E_p} \bar{x}_e$ for all $p \in P_s$. It follows:

$$\begin{aligned}
\sum_{p\in P_s} \bar{y}_{sp} &= \sum_{p\in P_s} \min\left\{1, \sum_{e\in E_p} \bar{x}_e\right\} \\
&= \sum_{p\in P_s} \sum_{e\in E_p} \bar{x}_e \\
&= \sum_{p\in P_s} \sum_{e\in E_p} b_{se}\bar{x}_e \\
&\geq \sum_{e\in E} b_{se}\bar{x}_e \geq 1,
\end{aligned}$$

where the last inequalities follow from the facts that each edge e with $b_{se} = 1$ may appear more than once in the double summation, and (3c) holds for $\bar{\mathbf{x}}$, respectively. Therefore, (4d) is also fulfilled in this case, which completes the proof. □

Note that the above proof does not hold for the general-demand case, because we cannot exploit $b_{sp} = b_{se}$ for all $e \in E_p$ and all $s \in S$.

5. Computational Results

In what follows, we describe the generation of the benchmark instances used in the following experiments, we compare the linear relaxations of formulations HI, EG, and RG, we report results for improved formulations in which only some (promising) parts of the original state spaces are replaced by subnetworks, and we finally compare the best formulations as integer programs.

5.1. Benchmark Instances

The instances of the SPRP-SS are generated as described in detail in Heßler and Irnich (2024) and Lüke et al. (2024). An instance is characterized by a combination of (m, C, a, α), where m denotes the number of aisles, C the number of locations per aisle, a the number of different articles to be collected, and α the scatter factor. The scatter factor describes the average number of pick positions at which an article is stored in the warehouse. We assume a classical single-block rectangular warehouse with $m = 10$ aisles and $C = 50$ locations per aisle. Additionally, we vary the number of different articles that need to be collected, i.e., $a \in \{30, 50, 100, 200\}$.

For the generation of a pick list and for the assignment of articles to pick positions, we use the following approach that is realistic and simpler than what has been used in Lüke et al. (2024) where class-based storage policies were analyzed. First, we randomly assign a different pick position $p \in P$ to every article $s \in S$ defining a pair $(s, p) \in S \times P$. Second, another $(\alpha - 1)a$ pairs (s, p) are randomly drawn while making sure that no identical pairs are generated. This procedure ensures that each article $s \in S$ can be found at least once in the warehouse, and that it can be collected from α different positions on average. Due to randomness, some articles $s \in S$ can be found at more than α positions while others at fewer positions. For the following studies, we use a scatter factor of $\alpha = 5$.

In addition, we generate instances with unit demand, i.e., $q_s = 1$ for all $s \in S$, and instances with general demand. For the latter, we first draw a random supply $b_{sp} \in \{1, 2, 3\}$ for each pair (s, p). Demands q_s are then drawn randomly from $\{1, 2, \ldots, \min\{6, \sum_p b_{sp}\}\}$. In particular, all general-demand instances are feasible by construction.

In order to produce statistically significant results, we generate 100 instances per combination (m, C, a, α). In total, the benchmark set comprises $2 \cdot 4 \cdot 100 = 800$ instances (factor 2 for unit and general demand; 4 combinations (m, C, a, α)). The benchmark is available at https://logistik.bwl.uni-mainz.de/research/benchmarks/.

5.2. *Linear Relaxation Results*

We first analyze empirically the strength of the linear relaxation of the formulations HI, EG, and RG. Recall that, for a given instance of the SPRP-SS, $z_{\text{SPRP-SS}}$ is the objective value. A lower bound is given by the objective value of the linear relaxation of the respective formulation, in the following denoted by z_{LP}. The *optimality gap* is defined as $100 \cdot (z_{\text{SPRP-SS}} - z_{\text{LP}})/z_{\text{SPRP-SS}}$, and it describes the strength of a formulation.

Table 1: Optimality gap

Formulation		general demand					unit demand				
	DCCs for	$a = 30$	50	100	200	avg.	$a = 30$	50	100	200	avg.
HI	O	4.83	3.47	1.71	0.33	2.58	6.30	8.31	8.60	7.76	7.74
EG	O, UA, UP	8.37	8.83	7.88	5.63	7.68	6.39	8.41	8.63	7.76	7.80
RG$^{(3)}$	O	*n.a.*	*n.a.*	*n.a.*	*n.a.*	*n.a.*	6.40	8.42	8.63	7.76	7.80
RG$^{(4)}$	O	5.28	3.69	1.82	0.35	2.79	12.21	14.03	13.19	11.09	12.63

Table 1 shows average optimality gaps (per group of 100 instances) for the formulations HI, EG, RG$^{(3)}$, and RG$^{(4)}$. Gaps are not available ("*n.a.*") for formulation RG$^{(3)}$ and general demand, since the model is not valid in this case. The linear relaxation of formulation HI always achieves the smallest gaps. This is not only true on average, but also for each single SPRP-SS instance and in general. This results from the fact that every feasible flow $\bar{\mathbf{x}} = (\bar{x}_e) \in [0,1]^E$ of the LP-model HI can be translated into a feasible flow of the other LP formulations. We refrain from a formal proof of this statement as the translation is trivial.

For the unit-demand case, we can observe the direct consequences of Property 6. Indeed, formulation RG$^{(3)}$ can be strictly stronger than formulation RG$^{(4)}$ in the unit-demand case. (The large difference can be explained by the fact that for an edge e that covers two or more positions in which an article s is stored the relation $b_{se} = 1 < \sum_{p \in P: e \in E_p} b_{sp}$ holds. Note that b_{se} is used in model (3) and b_{sp} in model (4).) Since formulation RG$^{(4)}$ is composed of more variables and constraints, it is strictly dominated. As a consequence, we will only consider the stronger formulation RG$^{(3)}$ for SPRP-SS with unit demand. Since model (4) is the only choice for general demand and the RG network, we can omit the superscript and refer to *formulation RG* in the following (to lighten the notation).

Comparing formulations EG and RG$^{(4)}$, we see from Table 1 that neither formulation dominates the other with regard to the linear relaxation. Average optimality gaps can differ substantially here, see $a = 100$ with gaps of 7.88 and 1.82 percent for general demand and gaps of 8.63 and 13.19 percent for unit demand, respectively. In contrast, for the unit-demand SPRP-SS instances that we consider, the average optimality gaps of formulations EG and RG$^{(3)}$ are very close.

5.3. *Vertex Factor*

The replacement of parallel edges of the original network by subnetworks of types EG and RG is reasonable only if the size of the new network is smaller. Indeed, less edges in the resulting EG and RG networks imply less variables in the resulting formulations EG and RG, respectively. However, the subnetworks introduce additional vertices v_{hi} or w_{hi}, one for each pick position or NEG, see Sections 4.1 and 4.2. Additional vertices require additional flow conservation constraints (3b) and (4b). Therefore, we propose to fine-control the replacement process by deciding, for each set of parallel edges separately, whether the replacement is performed or the parallel edges of the original network are kept. In this subsection, we will computationally analyze different options of controlling the replacement process.

To this end, for a subset of parallel edges under consideration, let k^{orig} and k^{new} denote the number of edges in the original and the new network. In addition, let ℓ_+^{new} denote the number of additional vertices v_{hi} or w_{hi} that would have to be included if the new replacement were performed. We use the condition

$$k^{\text{new}} + f\ell_+^{\text{new}} < k^{\text{orig}}$$

to decide on the replacement, where the *vertex factor* (VF) f estimates how strong the impact of the number ℓ_+^{new} of additional vertices is. Furthermore, we use the alternative conditions $k^{\text{new}} + (\ell_+^{\text{new}})^2 < k^{\text{orig}}$ (denoted as *quadratic*) and $k^{\text{new}} + (\ell_+^{\text{new}})^3 < k^{\text{orig}}$ (*cubic*) to simulate a very high impact of the additional vertices.

Table 2: Average Percentage of Edge Reduction, Replacements depending on VF f or rules *quadratic* and *cubic*.

Formulation		general demand					unit demand				
	VF f	$a = 30$	50	100	200	avg.	$a = 30$	50	100	200	avg.
EG	0	34.36	45.67	53.94	47.80	45.44	47.67	63.02	75.65	81.05	66.85
	1	32.28	44.12	52.78	45.79	43.74	47.15	62.99	75.64	81.05	66.71
	2	26.69	39.28	48.94	39.66	38.64	44.42	62.69	75.61	81.05	65.94
	3	19.38	33.20	43.01	31.08	31.67	36.97	61.80	75.55	81.04	63.84
	5	5.01	20.47	28.54	18.61	18.16	14.32	55.79	74.34	80.68	56.28
	10	0.00	0.61	8.91	2.62	3.03	0.00	3.54	65.61	72.79	35.48
RG	0	48.66	59.17	66.53	62.34	59.17	59.96	72.21	81.97	86.13	75.07
	1	48.52	59.10	66.51	62.31	59.11	59.95	72.21	81.97	86.13	75.06
	2	48.16	58.94	66.43	62.19	58.93	59.94	72.21	81.97	86.13	75.06
	3	47.69	58.74	66.33	62.03	58.70	59.92	72.21	81.97	86.13	75.06
	5	46.40	57.85	65.92	61.06	57.81	59.88	72.19	81.97	86.13	75.04
	10	42.09	55.13	63.70	56.36	54.32	59.41	72.17	81.97	86.13	74.92
	quadratic	47.50	58.14	65.29	57.66	57.15	59.94	72.20	81.97	86.13	75.06
	cubic	44.64	52.76	55.22	30.62	45.81	59.88	72.18	81.96	86.12	75.03

Table 2 shows the average percentage of edge reduction, i.e., $100 \cdot (1 - (|E| - |E^{\text{new}}|)/|E|)$, where E^{new} is the set of edges of the resulting network EG or RG. As before, the results are grouped by network type (EG or RG), the number a of different articles to pick, and the replacement condition used. Please note that the replacement rules *quadratic* and *cubic* did not significantly change the original network when subnetworks of type EG were tested. As a consequence, we do not report results for EG combined with *quadratic* and *cubic*. The most important results are:

(i) The reduction is stronger for the subnetworks RG than for the subnetworks EG.

(ii) Reductions increase with a, with the exception of general demand and $a = 200$, where one can observe a decline.

(iii) For the subnetworks EG, an increasing VF f strongly impacts the percentage reduction.

(iv) For the subnetworks RG, the respective reduction is moderately decreasing with an increasing VF f.

Table 3 uses the identical grouping and shows the average computation times of solving the formulations EG and RG, respectively. Compared to the percentage of arcs reduced (Table 2), the computation times are not directly related to the percentage reduction. This is a strong hint that also the number of vertices is important. However, none of the replacement rules and values of the VF f is a clear winner. For general demand, it seems that for smaller $a = 30$ and 50 the rule *quadratic* or a large VF f are advantageous for both types of networks EG and RG, while for larger $a = 100$ and 200 a smaller VF f leads to short computation times. For unit demand and network EG, it is difficult to give a good recommendation for the choice of VF f. In contrast, for unit demand and network RG, computation times are almost independent of the choice of VF f. Summarizing, Table 3 shows that using a small vertex factor between 0 and 2 often leads to the smallest times (over all replacement rules). As a consequence, we will use the replacement rule based on vertex factor $f = 1$ for the final experiments.

Finally, Table 3 shows that average computation times for subnetworks RG are always smaller than the corresponding times for subnetworks EG. Especially for general demand and $a = 100$ and 200, differences are substantial. We attribute the better performance of formulation RG in these cases of general demand

Table 3: Average Computation Times [in milliseconds], Replacements depending on VF f or rules *quadratic* and *cubic*.

Formulation		general demand					unit demand				
	VF f	$a = 30$	50	100	200	avg.	$a = 30$	50	100	200	avg.
EG	0	238	474	977	1158	712	199	456	1100	2197	988
	1	222	437	971	1259	722	205	452	1099	2105	965
	2	220	405	907	1397	732	198	452	1092	2195	984
	3	218	378	877	1553	756	210	441	1092	2190	983
	5	213	397	883	1685	795	245	469	1116	2104	984
	10	204	397	894	1884	845	178	673	1448	2839	1284
RG	0	191	248	375	596	353	160	347	827	1553	722
	1	189	248	376	600	353	160	349	830	1552	723
	2	188	244	372	603	352	158	348	829	1555	722
	3	188	246	377	614	356	159	349	829	1554	723
	5	188	249	376	644	364	159	348	830	1557	724
	10	189	271	401	776	409	164	349	830	1556	725
	quadratic	179	244	374	748	387	160	347	828	1553	722
	cubic	184	279	502	1404	592	160	348	829	1557	724

to the stronger linear relaxation which leads to larger search trees to be worked through by the MIP solver, see Table 1.

5.4. Integer Solutions

In the final computational experiments, we compare all three formulations HI, EG, and RG using the vertex factor $f = 1$ for the latter two. For formulations HI and RG, it is still not clear whether redundant DCCs for articles of type UA and UP should be used. To analyze this, we use three settings where DCCs are included either only for articles of type O, or of types O and UA, or for all types. Table 4 shows the average computation times of the MIP solver for the seven formulations. The results can be summarized as follows:

Table 4: Average Computation Times [in milliseconds], Final Setup with VF $f = 1$

Formulation		general demand					unit demand				
	DCCs for	$a = 30$	50	100	200	avg.	$a = 30$	50	100	200	avg.
HI	O	212	391	933	1933	867	183	632	2638	8165	2905
	O, UA	216	402	996	2048	916	182	638	2646	8986	3113
	O, UA, UP	213	400	957	1997	892	182	621	2574	7573	2737
EG	O, UA, UP	222	437	971	1259	722	205	452	1099	2105	965
	speedup	·0.90	·0.93	·1.13	·1.53	·1.10	·0.94	·1.36	·2.24	·3.40	·1.77
RG	O	189	248	376	600	353	160	349	830	1552	723
	O, UA	189	247	378	609	356	156	342	840	1599	734
	O, UA, UP	190	245	365	599	350	155	338	808	1426	682
	speedup	·1.02	·1.55	·2.56	·3.18	·1.89	·1.22	·1.84	·3.11	·4.87	·2.42

Note: Speedups are computed as geometric means of ratios of computation times relative to formulation HI with all DCCs (O, UA, UP).

(i) For general and unit demand, formulation RG is always solved fastest compared to formulations HI and EG. This is true for all values a and in the average (see column *avg.*).

(ii) Adding redundant DCCs (for articles of type UA and UP) accelerates the solution process for both formulation HI and formulation RG in the unit-demand case. Computation times for the general-demand case are not impacted significantly by the choice of DCCs.

Additionally, we compare the three formulations using the setting with all DCCs (for articles of types O, UA, and UP) by computing a *speedup* factor. The baseline is always formulation HI which represents the status quo. Thus, Table 4 presents speedup factors for formulations EG and RG, which are computed as geometric means of the ratios $t_{HI}(I)/t_{EG}(I)$ and $t_{HI}(I)/t_{RG}(I)$ where $t_{HI}(I)$, $t_{EG}(I)$, and $t_{RG}(I)$ are the computation times of the MIP solver using the respective formulation for solving SPRP-SS instance I. Any speedup value greater than one indicates that the new formulation is superior to the status quo for the subgroup of instances.

These numbers clearly indicate that already formulation EG can be advantageous compared to formulation HI. The speedup for formulation RG is even higher with an overall speedup factor between 1.02 and 3.18 for general demand and between 1.22 and 4.87 for unit demand. Notably, speedups are higher for larger instances with more articles to be collected, i.e., for larger values of a.

6. Conclusions

In this paper, we considered the SPRP-SS and its out-of-the-box solution with a MIP model and MIP solver. The approach we take is that of Heßler and Irnich (2024) who first build an extended network (inspired by Ratliff and Rosenthal's dynamic program) and then solve a shortest-path problem with additional DCCs with a MIP solver. This shortest-path network contains parallel edges representing alternative traversal actions of type `gap`, `bottom`, and `top` within the aisle under consideration. The two new formulations EG and RG, which have been developed here, replace sets of parallel edges of the aforementioned types by generally smaller subnetworks. The acronyms signify the possibility to enlarge `gap` (EG) actions or to reduce `gap` (RG) actions. To be more precise, if a path through the subnetwork can represent the action $\texttt{gap}(h, i)$, then it can also represent gaps of type $\texttt{gap}(h', i')$ with $h' \leq h$ or $i' \geq i$ in the subnetwork EG. For the subnetwork RG, the reduction of the action $\texttt{gap}(h, i)$ into $\texttt{gap}(h', i')$ is enabled where $h' \geq h$ or $i' \leq i$.

We analyzed the resulting formulations EG and RG and derived several theoretical properties of the formulations. To accommodate general demand, modifications were required to the structure of the MIP model for subnetworks EG, where we could still ensure that the resulting model remains linear. In particular, we showed that all new MIP models are linear in the number m of aisles and in the number n of possible pick positions. In contrast, the MIP model of Heßler and Irnich (2024) is quadratic in n.

We studied the performance of the MIP solver-based approach with the new formulations in three types of computational experiments, in which 800 SPRP-SS instances were solved with up to $a = 200$ articles. First, the linear relaxations of the new formulations EG and RG have similar integrality gaps in case of unit demand, but they are weaker than that of the formulation of Heßler and Irnich (2024) in case of general demand. Second, the reduction in network size is largest for the new formulation RG, where typically more than two-thirds of the edges can be reduced. These reductions translate to improved computation times of the MIP solver. Third, for a wide range of vertex factors, the simplest replacement rule (we use $f = 1$, i.e., we perform the replacement whenever the resulting network has at least one edge less) was identified working reasonably well, leading to substantially smaller MIP computation times for formulation EG and especially for formulation RG compared to formulation HI. Overall, compared to the MIP of Heßler and Irnich (2024) the average speedup is by the factor of 1.89 for general demand and of 2.42 for unit demand for the SPRP-SS instances in the testbed. For the largest instances with $a = 200$ articles, average speedups reach the factors of 3.18 and 4.87, respectively. In absolute numbers, average computation times, even for these largest instances, are below two seconds.

The new modeling approach may help to develop superior models for integrated problems in warehouse operations management, especially those where many instances of a problem similar to the SPRP-SS must be solved as subproblems multiple times.

Acknowledgement

Parts of this research were supported by Deutsche Forschungsgemeinschaft (DFG) under grant IR 122/13 of project 555303283.

References

David L. Applegate, Robert E. Bixby, Vašek Chvatal, and William J. Cook. Concorde-03.12.19. Website, 2003. https://www.math.uwaterloo.ca/tsp/concorde/index.html.

John J. Bartholdi, III and Steven T. Hackman. *Warehouse & Distribution Science*. Atlanta, GA 30332-0205 USA, release 0.98.1 edition, 2019. Online only, https://www.warehouse-science.com/book/editions/wh-sci-0.98.1.pdf.

Nils Boysen, René de Koster, and Felix Weidinger. Warehousing in the e-commerce era: A survey. *European Journal of Operational Research*, 277(2):396–411, 2019. doi:10.1016/j.ejor.2018.08.023.

Melh Çelk and Haldun Süral. Order picking under random and turnover-based storage policies in fishbone aisle warehouses. *IIE Transactions*, 46(3):283–300, 2014. doi:10.1080/0740817x.2013.768871.

Richard L. Daniels, Jeffrey L. Rummel, and Robert Schantz. A model for warehouse order picking. *European Journal of Operational Research*, 105(1):1–17, 1998. doi:10.1016/S0377-2217(97)00043-X.

Matteo Fischetti, Juan-Jose Salazar-Gonzalez, and Paolo Toth. The generalized traveling salesman and orienteering problems. In G. Gutin and A.P. Punnen, editors, *The Traveling Salesman Problem and Its Variations*, volume 12 of *Combinatorial Optimization*, chapter 13, pages 609–662. Kluwer, Dordrecht, 2002. doi:10.1007/0-306-48213-4_13.

Dominik Goeke and Michael Schneider. Modeling single-picker routing problems in classical and modern warehouses. *INFORMS Journal on Computing*, 33(2):436–451, 2021. doi:10.1287/ijoc.2020.1040.

Randolph W. Hall. Distance approximations for routing manual pickers in a warehouse. *IIE Transactions*, 25(4):76–87, 1993. doi:10.1080/07408179308964306.

Mustapha Haouassi, Yannick Kergosien, Jorge E. Mendoza, and Louis-Martin Rousseau. The picker routing problem in mixed-shelves, multi-block warehouses. *International Journal of Production Research*, pages 1–22, 2024. doi:10.1080/00207543.2024.2374845.

Keld Helsgaun. An effective implementation of the Lin–Kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130, 2000. doi:10.1016/s0377-2217(99)00284-2.

Katrin Heßler and Stefan Irnich. A note on the linearity of Ratliff and Rosenthal's algorithm for optimal picker routing. *Operations Research Letters*, 50(2):155–159, 2022. doi:10.1016/j.orl.2022.01.014.

Katrin Heßler and Stefan Irnich. Exact solution of the single picker routing problem with scattered storage. *INFORMS Journal on Computing*, 2024. doi:10.1287/ijoc.2023.0075. Articles in Advance.

Imran Khan, Olaf Maurer, Julius Pätzold, Paweł Pszona, and Jan-David Salchow. Joint order selection, allocation, batching and picking for large scale warehouses. arXiv 2401.04563, 2024. https://arxiv.org/abs/2401.04563.

Laura Lüke, Katrin Heßler, and Stefan Irnich. The single picker routing problem with scattered storage: Modeling and evaluation of routing and storage policies. *OR Spectrum*, 46:909–951, 2024. doi:10.1007/s00291-024-00760-4.

Makusee Masae, Christoph H. Glock, and Eric H. Grosse. Order picker routing in warehouses: A systematic literature review. *International Journal of Production Economics*, 224:107564, 2020. doi:10.1016/j.ijpe.2019.107564.

C. E. Miller, Albert W. Tucker, and R. A. Zemlin. Integer programming formulations and traveling salesman problems. *Journal of Association for Computing Machinery*, 7:326–329, 1960. doi:10.1145/321043.321046.

Ömer Öztürkoğlu, Kevin R. Gue, and Russell D. Meller. Optimal unit-load warehouse designs for single-command operations. *IIE Transactions*, 44(6):459–475, 2012. doi:10.1080/0740817x.2011.636793.

Lucie Pansart, Nicolas Catusse, and Hadrien Cambazard. Exact algorithms for the order picking problem. *Computers & Operations Research*, 100:117–127, 2018. doi:10.1016/j.cor.2018.07.002.

Charles G. Petersen. An evaluation of order picking routeing policies. *International Journal of Operations & Production Management*, 17(11):1098–1111, 1997. doi:10.1108/01443579710177860.

H. Donald Ratliff and Arnon S. Rosenthal. Order-picking in a rectangular warehouse: A solvable case of the traveling salesman problem. *Operations Research*, 31(3):507–521, 1983. doi:10.1287/opre.31.3.507.

Kees Jan Roodbergen and René de Koster. Routing methods for warehouses with multiple cross aisles. *International Journal of Production Research*, 39(9):1865–1883, 2001. doi:10.1080/00207540110028128.

Kashi N. Singh and Dirk L. van Oudheusden. A branch and bound algorithm for the traveling purchaser problem. *European Journal of operational research*, 97(3):571–579, 1997. doi:10.1016/S0377-2217(96)00313-X.

Yixuan Su, Xi Zhu, Jinlong Yuan, Kok Lay Teo, Meixia Li, and Chunfa Li. An extensible multi-block layout warehouse routing optimization model. *European Journal of Operational Research*, 305(1):222–239, 2023. doi:10.1016/j.ejor.2022.05.045.

Felix Weidinger. Picker routing in rectangular mixed shelves warehouses. *Computers & Operations Research*, 95:139–150, 2018. doi:10.1016/j.cor.2018.03.012.

Felix Weidinger, Nils Boysen, and Michael Schneider. Picker routing in the mixed-shelves warehouses of e-commerce retailers. *European Journal of Operational Research*, 274(2):501–515, 2019. doi:10.1016/j.ejor.2018.10.021.

Constantin Wildt, Felix Weidinger, and Nils Boysen. Picker routing in scattered storage warehouses: an evaluation of solution methods based on TSP transformations. *OR Spectrum*, 2024. doi:10.1007/s00291-024-00780-0.