OR Spectrum (2023) 45:1071–1111
https://doi.org/10.1007/s00291-023-00728-w

ORIGINAL ARTICLE

A new modeling approach for the unrestricted block relocation problem

Alf Kimms[1] · Fabian Wilschewski[1]

Received: 4 July 2022 / Accepted: 20 June 2023 / Published online: 7 July 2023

Abstract

We consider the block relocation problem (BRP), a combinatorial optimization problem that may arise in storage systems where items are organized in stacks. The objective is to retrieve all items in a predefined order with a minimal number of relocations. It can be distinguished between a restricted and an unrestricted version of the BRP. While in the restricted BRP (R-BRP) only relocations of items located above the item to be retrieved next are permitted, in the unrestricted BRP (U-BRP) all possible relocations are allowed. Existing exact methods concerning the BRP are frequently search-based methods which appear to be very effective. Nevertheless, recent literature concerning the R-BRP has shown that model-based methods can be competitive and therefore should also be taken into consideration. In this paper, we propose a new model-based approach for the U-BRP. It eliminates the fact that the number of variables is increasing with the number of necessary relocations; a disadvantage most mathematical models for the U-BRP have in common.

Keywords Combinatorial optimization · Block relocation problem · Mixed integer linear model · Row generation

1 Introduction and problem description

In a storage system, the space available for storing items is scarce. Due to that fact, items are often stored in stacks, i.e., on top of each other. For example, this is a common approach for storing containers in a container terminal. Due to technical conditions, items are only accessible from above, e.g., via rail mounted gantry cranes. Consequently, an item can be blocked by other items stored above it. If an item to be

✉ Alf Kimms
alf.kimms@uni-due.de

Fabian Wilschewski
fabian.wilschewski@uni-due.de

1 Mercator School of Management, University of Duisburg-Essen, Lotharstr. 65, 47057 Duisburg, Germany

retrieved is blocked, repositioning moves are necessary. Since repositioning moves may be very time consuming, minimizing the total number of these moves can have crucial impact on the efficiency of such a storing system.

An optimization problem that may arise in a storing system like this is the well-known block relocation problem (BRP) which can be stated as follows: There is a given number N of items with labels 1 to N piled up in W last-in-first-out stacks with limited height H. The N items have to be retrieved in a predefined order $1, \dots, N$ until all stacks are cleared. The item to be retrieved next is called *target item* and the stack in which it is located is called *target stack*. If a target item is not the topmost of its stack, all items located above it must be relocated to other stacks first. So there is a distinction between two types of *moves*: *relocations* and *retrievals*. A *relocation* moves an item from its current position to another position within the stacking area. The new position is always on top of another (possibly empty) stack. A *retrieval* removes an item from the stacking area. The objective is to retrieve all items in the predefined order with a minimal number of relocations. Note that the number of retrievals is a constant since it always equals the number N of items and therefore does not need to be taken into consideration within the objective.

In the top left corner of Fig. 1 (at $c = 0$), a stacking area with an initial configuration consisting of $W = 3$ stacks (stack 3 is an empty stack) with a limited height $H = 3$ is depicted. A position within the two-dimensional stacking area is denoted as a *slot* (i, j) which is a 2–tuple consisting of a stack $i \in \{1, \dots, W\}$ and a tier $j \in \{1, \dots, H\}$, e.g., item 1 is located in *slot* (1, 2). A configuration (1) each slot is occupied by at most one item and (2) each item is located on top of another item or on the floor, i.e., there are no floating items.

A solution of the BRP is a sequence of moves (relocations and retrievals). Each move of such a sequence leads to a transition from a configuration $c - 1$ to a configuration c. The transition which results in configuration c is called the c-th transition. A solution is feasible if (1) the resulting configurations after each single move within the sequence are feasible, (2) the retrieval order is correct and (3) the stacking area is cleared in the end.

Figure 1 presents an optimal solution of the unrestricted BRP (U-BRP) for an exemplary initial configuration. Let the occurring configurations be numbered by

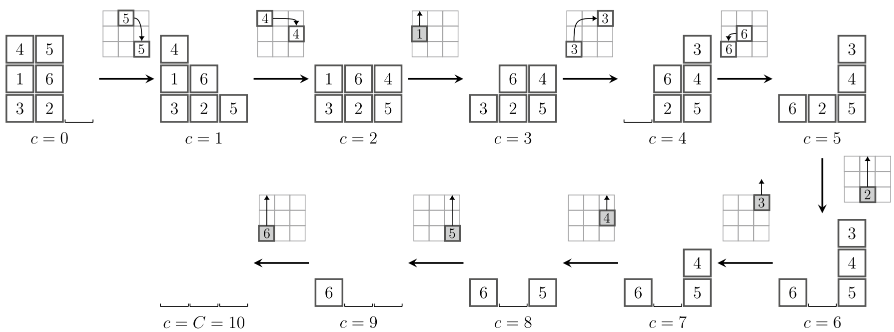

Fig. 1 Optimal solution of the U-BRP with objective function value 4 (the total number of relocations)

$c = 0, \ldots, C$. After relocating item 5 from *slot* (2, 3) to *slot* (3, 1) within the first transition, item 2 is only blocked by one item, namely item 6. Within the second transition, item 4 can be relocated onto item 5 without creating a further blockage. Since item 1 is the topmost of its stack in configuration $c = 2$ it can be retrieved within the third transition. After two more relocations the configuration $c = 5$ is free of blockages and hence all items can be retrieved without further relocations. Therefore, an optimal solution of the U-BRP contains four relocations to clear the stacking area. It can be observed that the first relocation does not originate in the target stack. This opportunity of presorting items characterizes the U-BRP and can be advantageous with respect to the objective of minimizing the total number of relocations.

A second version of the BRP is the restricted BRP (R-BRP). It is based on an assumption proposed by Kim and Hong (2006) which prevents relocations of items not located in the target stack. This reduces the complexity of the problem since the decision on which item to relocate next is fixed. The only decision to make is determining the stack to which a relocation moves a blocking item. An optimal solution of the R-BRP for the identical example of Fig. 1 requires five relocations.

The main contribution of our paper is a new modeling approach for the U-BRP with a reduced number C of transitions (or configurations) that are considered. To the best of our knowledge, all mathematical model formulations for the U-BRP in the literature allow at most one relocation per transition. Thus, the size of these model formulations increases with the number of necessary moves. In our approach, the number C of transitions (or configurations) which need to be considered equals the number N of items, achieved by allowing multiple moves within each transition. We show that modifying a model formulation in such a manner may significantly increase its performance.

This paper is organized as follows: In Sect. 2, we give an overview of relevant literature. A mixed integer linear model formulation whose size is independent of the number of relocations required is presented in Sect. 3. In Sect. 4, a row generation framework is proposed to construct an optimal solution for the original problem U-BRP by means of the model formulation. Acceleration methods to speed up the row generation procedure are proposed in Sect. 5. In Sect. 6, the results of a computational study are presented which examines if a modeling approach may benefit from the reduction of the number of considered configurations. Finally, a short conclusion is given in Sect. 7.

2 Literature review

The BRP may arise in any storage system in which items are organized in last-in-first-out stacks, e.g., containers in a container terminal, pallets or steel plates in a warehouse, freight cars in a shunting yard etc. Due to the large and still increasing amount of cargo being shipped in containers via sea transportation, the field of container terminal logistics is apparently one of the most relevant areas. According to that, the BRP is also known as the container relocation problem (CRP). For further insights into container terminal logistics and operations research applications see, e.g., Steenken et al. (2004), Stahlbock and Voß (2008), Caserta et al. (2011) and Covic (2019).

The BRP is an $\mathcal{NP}$-hard problem (see Caserta et al. (2012)) which belongs to the research field of storage problems. Storage problems can be classified into storage loading, unloading and premarshalling problems. Storage loading problems deal with incoming items which have to be assigned to locations within a storage area. Within storage unloading problems, in turn, items are retrieved and therefore leave the storage area. The BRP is a typical variant of a storage unloading problem where the storage area is organized in last-in-first-out stacks. Lastly, storage premarshalling problems deal with presorting operations without any items entering or leaving the storage area. Additionally, there exist combined problems, e.g., simultaneous planning of loading and unloading operations. For a detailed classification scheme of storage problems and a categorisation of existing literature see Lehnfeld and Knust (2014).

The BRP has been introduced by Kim and Hong (2006) in the restricted version. The authors propose an assumption called Assumption A1 which permits only relocations above the target item. They develop a heuristic based on a calculation of the “expected number of additional relocations” caused by executed moves as well as an exact branch–and–bound procedure. The Assumption A1 is widely used in the literature and therefore a large part of subsequent studies addresses the R-BRP. For the R-BRP, there exist several heuristics (see Bacci et al. 2019; Caserta et al. 2009, 2011, 2012; Jin et al. 2015; Jovanovic and Voß 2014; Jovanovic et al. 2019; Tang et al. 2015; Ting and Wu 2017; Ünlüyurt and Aydın 2012; Wan et al. 2009; Zhang et al. 2020) and exact methods (see Bacci et al. 2020; Expósito-Izquierdo et al. 2014, 2015; Ku and Arthanari 2016; Quispe et al. 2018; Tanaka and Takii 2016; Tanaka and Mizuno 2018; Tanaka and Voß 2022; Ünlüyurt and Aydın 2012; Zhang et al. 2010, 2020; Zhu et al. 2012). Some of the most effective exact methods for the R-BRP are search-based algorithms, like the branch–and–bound algorithms proposed by Tanaka and Mizuno (2018) or Zhang et al. (2020) as well as an iterative deepening A* (IDA*) algorithm proposed by Quispe et al. (2018). Nevertheless, Bacci et al. (2020) present a branch–and–cut algorithm and Tanaka and Voß (2022) propose an iterative algorithm based on binary linear models which both have proven to be very competitive model-based approaches for the R-BRP. In particular, Tanaka and Voß (2022) compared their model-based approach with the model-based approach of Bacci et al. (2020) and with three variants of one of the best-performing search-based approaches of Tanaka and Mizuno (2018) on 840 instances [test (CV) instances proposed by Caserta et al. (2011) and Caserta et al. (2012)]. The results show that the model-based approach of Tanaka and Voß (2022) outperforms all other investigated methods on these instances and that it is also the first algorithm which can optimally solve all instances of the dataset.

For the U-BRP, there exists a smaller number of heuristics (see Expósito-Izquierdo et al. 2014; Forster and Bortfeldt 2012; Jovanovic et al. 2019; Petering and Hussein 2013; Tricoire et al. 2018) and exact methods (see Expósito-Izquierdo et al. 2014; Jin and Tanaka 2023; Tanaka and Mizuno 2015, 2018; Tricoire et al. 2018; Zhu et al. 2012). Analogously to the literature concerning exact methods for the R-BRP, most existing exact methods for the U-BRP are search-based algorithms, like branch–and–bound algorithms (see Jin and Tanaka 2023; Tanaka and Mizuno 2015, 2018; Tricoire et al. 2018), an A* algorithm (see Expósito-Izquierdo et al. 2014) and an IDA* algorithm (see Zhu et al. 2012).

Furthermore, mathematical model formulations for the BRP have been developed. Kim and Hong (2006) and Caserta et al. (2011) present dynamic programming formulations for the R-BRP. Wan et al. (2009) develop a binary linear model called MRIP (Minimization of Reshuffles IP) for the R-BRP which is corrected and further improved by Tang et al. (2015). Caserta et al. (2012) propose another binary linear model for the R-BRP called BRP–II. For specific instances, the BRP–II falsely has no feasible solution which is corrected by Expósito-Izquierdo et al. (2015) and Zehendner et al. (2015). Furthermore, Zehendner et al. (2015) present a binary linear model called BRP–II–A which is an improved version of the BRP–II with a reduced number of binary variables. Zehendner and Feillet (2014) develop a reformulation of the BRP–II which is used within a branch–and–price framework. A further binary linear model for the R-BRP called CRP–I with an even lower number of binary variables than the BRP–II–A is proposed by Galle et al. (2018). The authors make use of a binary encoding of a configuration as proposed by Caserta et al. (2009). Bacci et al. (2020) develop a compact binary model for the R-BRP which contains logical constraints. Due to the complexity of these constraints, they are relaxed and the remaining model is embedded within a branch–and–cut framework. Most recently, Tanaka and Voß (2022) propose a binary linear model for the R-BRP which depends on sets of all possible relocation sequences for each individual item. Due to the exponential number of these sequences, the authors present a relaxed model which is embedded within an iterative algorithm.

For the U-BRP, Lee and Hsu (2007) formulate an integer model as a multicommodity flow problem. The authors actually address the premarshalling problem, but they propose model modifications which also make it applicable for the U-BRP. A binary linear model for the U-BRP called BRP–I is proposed by Caserta et al. (2012). Petering and Hussein (2013) develop a mixed binary linear model for the U-BRP called BRP–III which has a decreased number of binary variables in comparison with the BRP–I. A unified binary linear model which is applicable for the premarshalling problem, the R-BRP and the U-BRP, is proposed by de Melo da Silva et al. (2018). They present two variants of model formulations called BRP_{m1} and BRP_{m2}, whereas BRP_{m2} outperforms all existing model formulations for the U-BRP so far. Finally, Lu et al. (2020) develop a further unified binary linear model called BRP_{m3} which is, to the best of our knowledge, the state–of-the-art model formulation for the U-BRP.

A recent overview of existing literature concerning the BRP can be found in Lu et al. (2020). The authors propose a classification scheme which consists of 16 variants of the BRP. Furthermore, they give a recap of existing lower bounds on the number of relocations and their development in the literature as well as a proposal for a new and stronger lower bound.

3 Mathematical model formulation

In this section, we propose a mixed integer linear model called U-BRP$_{\text{master}}$. The notation and the structure of the model formulation is similar to the BRP–I proposed by Caserta et al. (2012). The BRP–I as well as the other existing mathematical model formulations for the U-BRP has in common that they permit at most one relocation per transition. This property makes it easy to prevent infeasible relocations.

As a major difference to these model formulations, we are allowing multiple relocations per transition. This provides the possibility to decrease the size of the model since the number of configurations which must be considered can be reduced. Even though there exist more efficient model formulations for the U-BRP, as presented in Sect. 2, we make use of the BRP–I as a basis since we can exploit some of its structural properties. Furthermore, it is very suitable for a reduction of the number of occurring configurations and therefore for an investigation if such a modification may involve benefits.

Note that within model formulations which ensure the configurations to be feasible after each individual move, the number C of occurring configurations needs to be an upper bound on the total number of moves (relocations and retrievals). In contrast to that, within the proposed U-BRP$_{\text{master}}$ the configurations are not ensured to be feasible after every move, but only after each retrieval. Therefore, the number C of configurations which must be considered is determined by the number of retrievals or items ($C = N$), respectively.

The transition resulting in configuration c contains the retrieval of item $n = c$, i.e., within the c-th transition the item c is the target item. In addition, several relocations may take place in a transition as well. Remember the optimal solution of the U-BRP for our example presented in Fig. 1. Figure 2 presents the same solution adapted to the framework of the U-BRP$_{\text{master}}$. Within the U-BRP$_{\text{master}}$, e.g., the first transition contains the retrieval of item 1 and additionally two relocations which results in the feasible configuration $c = 1$.

A quick recap of the parameters: W is the number of stacks, H is the maximal height of the stacks, N is the number of items and items are numbered $1, \dots, N$ which indicates the sequence in which items are to be retrieved. Additionally, there exist binary

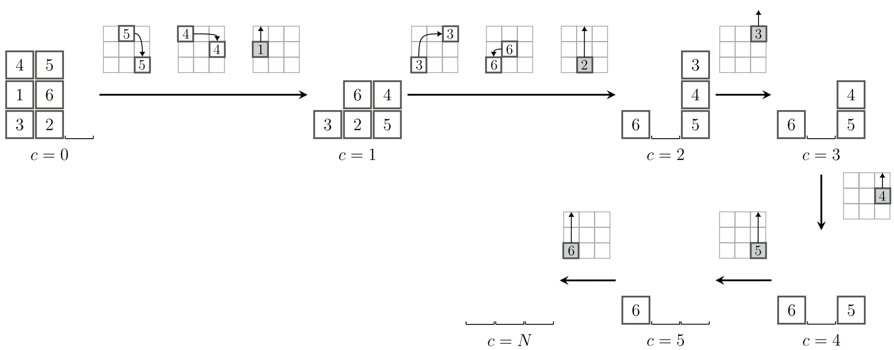

Fig. 2 A feasible solution of the U-BRP$_{\text{master}}$

parameters $\bar{v}_{nc}$ which take value 1 if item n has already been retrieved and therefore is not anymore located within the stacking area of configuration c (0 otherwise). Note that we know in advance within which (c-th) transition a specific item n is retrieved and therefore $\bar{v}_{nc}$ are not decision variables but parameters defined as follows:

$$\bar{v}_{nc} := \begin{cases} 0, & n = 2, \dots, N; c = 1, \dots, n-1 \\ 1, & n = 1, \dots, N; c = n, \dots, N \end{cases} \tag{1}$$

Furthermore, there are four types of decision variables in the U-BRP$_{\text{master}}$. The variables $\bar{b}_{ijnc}$ take value 1 if item n is located in *slot* (i, j) of configuration c (0 otherwise). Note that $\bar{b}_{ijn0}$ are parameters and define the initial configuration $c = 0$. The value of the variables $\bar{x}_{ijklnc}$ represent the number of how many times an item n is relocated from *slot* (i, j) of configuration $c - 1$ to *slot* (k, l) of configuration c (0 otherwise), i.e., within the c-th transition. Note that usually $\bar{x}_{ijklnc}$ takes values $\{0, 1\}$ only since it is very unlikely that an optimal solution contains an identical relocation of the same item more than once within a transition. Nevertheless, we do not exclude this opportunity in order not to falsely cut off such an optimal solution (see the Appendix 1 for a feasible solution with an $\bar{x}$–variable having the value 2). The binary variables $\bar{y}_{ijnc}$ take value 1 if item n is retrieved from *slot* (i, j) within the c-th transition (0 otherwise) and therefore item n is not anymore located within the stacking area of configuration c. The variables B_{ij} take value 1 if *slot* (i, j) contains an item within configuration $c = N - W$ which is blocking any of the items located below it (0 otherwise). Finally, the mathematical program can be stated as follows:

$$\text{U-BRP}_{\text{master}}: \quad \min \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{W} \sum_{l=1}^{H} \sum_{n=1}^{N} \sum_{c=1}^{N-W} \bar{x}_{ijklnc} + \sum_{i=1}^{W} \sum_{j=2}^{\min(H,W)} B_{ij} \tag{2}$$

s.t.

$$\sum_{i=1}^{W} \sum_{j=1}^{H} \bar{b}_{ijnc} + \bar{v}_{nc} = 1 \qquad n = 1, \dots, N; c = 1, \dots, N - W \tag{3}$$

$$\sum_{n=1}^{N} \bar{b}_{ijnc} \leq 1 \qquad i = 1, \dots, W; j = 1, \dots, H; c = 1, \dots, N - W \tag{4}$$

$$\sum_{n=1}^{N} \bar{b}_{ijnc} \geq \sum_{n=1}^{N} \bar{b}_{i(j+1)nc} \qquad i = 1, \dots, W; j = 1, \dots, H - 1; \; c = 1, \dots, N - W \tag{5}$$

$$\bar{b}_{ijnc} = \bar{b}_{ijn(c-1)} + \sum_{k=1}^{W} \sum_{l=1}^{H} \bar{x}_{klijnc} - \sum_{k=1}^{W} \sum_{l=1}^{H} \bar{x}_{ijklnc} - \bar{y}_{ijnc} \qquad i = 1, \dots, W; j = 1, \dots, H; n = 1, \dots, N; c = 1, \dots, N - W \tag{6}$$

$$\bar{v}_{nc} = \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{c'=1}^{c} \bar{y}_{ijnc'} \qquad n = 1, \dots, N; c = 1, \dots, N - W \tag{7}$$

$$\bar{x}_{ijilnc} = 0 \qquad i = 1, \dots, W; j, l = 1, \dots, H; n = 1, \dots, N; \\ c = 1, \dots, N - W \tag{8}$$

$$\bar{b}_{ijn(N-W)} + \sum_{n'=1}^{n-1} \bar{b}_{ij'n'(N-W)} - 1 \le B_{ij} \qquad i = 1, \dots, W; \\ j = 2, \dots, \min(H, W); j' = 1, \dots, j - 1; n = N - W + 2, \dots, N \tag{9}$$

$$\bar{b}_{ijnc} \ge 0 \qquad i = 1, \dots, W; j = 1, \dots, H; n = 1, \dots, N; c = 1, \dots, N - W \tag{10}$$

$$\bar{x}_{ijklnc} \in \mathbb{N}_0 \qquad i, k = 1, \dots, W; j, l = 1, \dots, H; n = 1, \dots, N; \\ c = 1, \dots, N - W \tag{11}$$

$$\bar{y}_{ijnc} \in \{0, 1\} \qquad i = 1, \dots, W; j = 1, \dots, H; n = 1, \dots, N; \\ c = 1, \dots, N - W \tag{12}$$

$$B_{ij} \ge 0 \qquad i = 1, \dots, W; j = 2, \dots, \min(H, W) \tag{13}$$

Within the U-BRP$_{\text{master}}$, the configuration $c = N - W$ is of special importance. In configuration $c = N - W$, the number of remaining items equals the number W of stacks. Thus, each blocking item can be relocated to an individual stack and the problem becomes trivial for all succeeding transitions. For these transitions $N - W + 1, \dots, N$, the number of relocations required equals the number of blocking items and therefore it is not necessary to calculate the moves of these transitions. As a result, most decision variables and constraints are not defined for $c > N - W$. This insight is based on a model formulation for the R-BRP called CRP–I proposed by Galle et al. (2018). For illustration purposes, most figures depict all transitions $1, \dots, N$ and its contained moves, even though the moves of the transitions $N - W + 1, \dots, N$ are not explicitly calculated within the U-BRP$_{\text{master}}$.

According to that, the objective (2) minimizes the sum of the number of relocations within the transitions $1, \dots, N - W$ and the number of blocking items within configuration $c = N - W$. Note that the number $\min(H, W)$ is the maximal height a stack can have within the configuration $c = N - W$ so that blocking items cannot be located on a tier $j > \min(H, W)$. Constraints (3) state that in a configuration an item can either be located inside or outside of the stacking area, not both. Constraints (4) make sure that a *slot* (i, j) cannot be occupied by more than one item within a configuration. Constraints (5) prevent an empty *slot* (i, j) within a configuration if an item is located above it in *slot* $(i, j + 1)$, i.e., there must not be floating items. Constraints (6) are "flow balancing constraints" which transfer the correct positions of all items from a configuration $c-1$ to its consecutive configuration c considering all possibly performed relocations and retrievals. Note that $\bar{b}_{ijn0}$ are binary parameters and define the initial configuration of the stacking area. Constraints (7) ensure that an item which is not anymore located within the stacking area of configuration c ($\bar{v}_{nc}=1$) must be retrieved in a preceding transition $c'=1,\ldots,c$ ($\bar{y}_{ijnc'}=1$). Additionally, constraints (8) are applied to the U-BRP$_{\text{master}}$ to prohibit relocations within one stack since such relocations are never feasible with respect to the original problem. Actually, the variables $\bar{x}_{ijklnc}$ with $i=k$ can be omitted from the U-BRP$_{\text{master}}$, but for a better readability and to go in line with the remainder of this paper, we just state this by (8). Constraints (9) are introduced for counting the number of blocking items by means of the variables B_{ij}. If there is an item n located in *slot* (i,j) of configuration $c=N-W$ and there is an item $n'<n$ located in any of the slots below *slot* (i,j), then the corresponding variable B_{ij} is enforced to take the value 1. The domains of the decision variables are declared by (10)–(13). Note that even though the variables $\bar{b}_{ijnc}$ and B_{ij} are defined continuous, the model enforces these variables to take values $\{0,1\}$ only ($\bar{b}_{ijnc}$ enforced by (4) + (6), B_{ij} enforced by (2) + (9)).

A drawback of the U-BRPmaster is that it may deliver optimal solutions which are infeasible with respect to the original problem U-BRP. Remember that the U-BRP$_{\text{master}}$ does not contain a configuration after each move, i.e., there are omitted configurations and therefore not every individual move is checked for feasibility. This means that even though optimally solving the U-BRP$_{\text{master}}$ delivers a solution with feasible configurations only, there might be omitted configurations which would be infeasible.

Note that Fig. 2 shows a feasible but not an optimal solution of the U-BRP$_{\text{master}}$. An optimal solution of the U-BRP$_{\text{master}}$ for the same instance is presented in Fig. 3. The first and second transitions in Fig. 3 include two and three moves, respectively. The first transition, for example, includes one relocation ($\bar{x}^*_{231251}=1$) and one retrieval ($\bar{y}^*_{1211}=1$) leading to the feasible configuration $c=1$, but the U-BRP$_{\text{master}}$ does not deliver information about the order of multiple moves within a transition. It is obvious that a solution must lead to a feasible configuration after each single move to be feasible for the original problem U-BRP. In our small example it is easy to see that the moves of the first transition cannot be executed in any order. If the

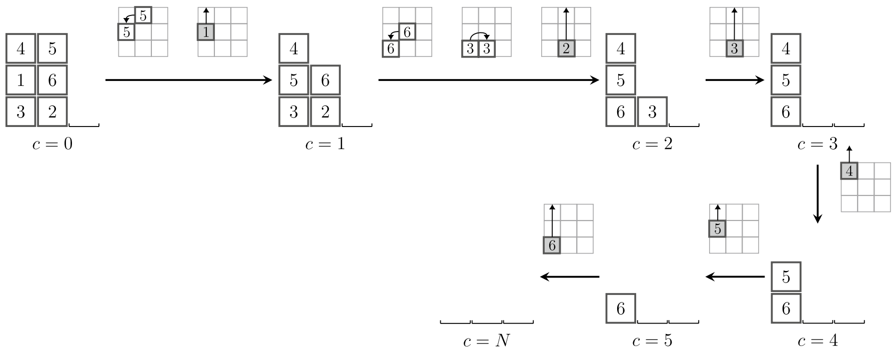

Fig. 3 An optimal solution of the U-BRP$_{\text{master}}$ with objective function value 3

relocation of item 5 is performed first, the result would be a configuration with *slot* (1, 2) being occupied by the two items 1 and 5 which is infeasible. On the other hand, retrieving item 1 first would lead to a configuration including floating items which is also infeasible. For the second transition, there does not exist any feasible order with respect to the original problem U-BRP, either.

To guarantee feasibility (and optimality) of the solutions, the U-BRP$_{\text{master}}$ is embedded within a row generation framework. Within this procedure, for each transition including multiple moves that cannot be executed in any order, a constraint is generated and added to the U-BRP$_{\text{master}}$ to prohibit a specific combination of moves. This procedure is presented in the next section.

4 A row generation framework

In the following, we propose a row generation framework to construct an optimal solution for the original problem U-BRP by means of the model formulation U-BRP$_{\text{master}}$. The idea is to generate and add constraints to the U-BRP$_{\text{master}}$ until the optimal solution of the U-BRP$_{\text{master}}$ is feasible for the original problem U-BRP. Note that every optimal solution of the U-BRP is feasible for the U-BRP$_{\text{master}}$, but not vice versa. If a specific combination of moves cannot be part of a feasible solution of the U-BRP, then this combination of moves is prohibited within the U-BRP$_{\text{master}}$ by an additional constraint. Thus, an optimal solution of the original problem U-BRP is never cut off by a generated constraint.

Given an optimal solution of the U-BRP$_{\text{master}}$, for each transition $c = 1, \dots, N - W$ a subproblem is to be solved. In each subproblem, it must be examined if there is any order for the corresponding moves leading to feasible configurations after each single move. Thus, if a transition contains only one move, there is no subproblem to be solved. Remember that within the c-th transition, the item c is the target item and is therefore retrieved. So if there is only one move within a transition, this move is a retrieval. Remember that the transitions $N - W + 1, \dots, N$ are not contained within the U-BRP$_{\text{master}}$ and therefore do not need to be considered.

In the next Sect. 4.1, the general outline of the row generation procedure is presented and illustrated by means of a small example and due to the small size of the example the subproblems are solved by just looking closely. Actually, within the implementation of the whole procedure, the subproblems are solved with the help of a model formulation. This model formulation for the subproblems (separation problems) to identify infeasible combinations of moves is given in Sect. 4.2.

4.1 General outline

Recall the optimal solution of the U-BRP$_{\text{master}}$ of our example presented in Fig. 3. The first subproblem corresponds to the first transition, i.e., the transition from configuration $c = 0$ to $c = 1$. As already mentioned in Sect. 3, there exists no order for the moves of the first transition ($\bar{x}^*_{231251} = 1$ and $\bar{y}^*_{1211} = 1$) which leads to a feasible configuration after each single move. Thus, this combination of moves cannot be part of a feasible solution of the original problem U-BRP. To prohibit this combination, the model U-BRP$_{\text{master}}$ is extended by an additional constraint.

Let γ_c be the number of moves within the c-th transition of an optimal solution of the U-BRP$_{\text{master}}$, formally defined as

$$\gamma_c := \sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{k=1}^{W}\sum_{l=1}^{H}\sum_{n=1}^{N}\bar{x}^*_{ijklnc} + \sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{n=1}^{N}\bar{y}^*_{ijnc} \qquad c = 1,\dots,N-W \tag{14}$$

Furthermore, let $X := \{(i,j,k,l,n) \mid i,k = 1,\dots,W; j,l = 1,\dots,H; n = 1,\dots,N\}$ be the set of 5-tuples (i, j, k, l, n) corresponding to all relocations that may or may not take place within a transition and let $Y := \{(i,j,n) \mid i = 1,\dots,W; j = 1,\dots,H; n = 1,\dots,N\}$ be the set of 3-tuples (i, j, n) corresponding to all retrievals that may or may not take place within a transition.

For all $c = 1,\dots,N-W$, let $X^*_c := \{(i,j,k,l,n) \mid (i,j,k,l,n) \in X : \bar{x}^*_{ijklnc} \geq 1\}$ be the set of the 5-tuples (i, j, k, l, n) corresponding to the relocations contained in the c-th transition of the current optimal solution of the U-BRP$_{\text{master}}$. Analogously, for all $c = 1,\dots,N-W$, let $Y^*_c := \{(i,j,n) \mid (i,j,n) \in Y : \bar{y}^*_{ijnc} = 1\}$ be the set including the 3-tuple (i, j, n) corresponding to the retrieval within the c-th transition of the optimal solution and let $B^*_c := \{(i,j,n) \mid (i,j,n) \in Y : \bar{b}^*_{ijnc} = 1\}$ be the set of the 3-tuples (i, j, n) corresponding to the locations of the items within configuration c of the optimal solution.

For each transition $c = 1,\dots,N-W$ containing a combination of moves which cannot be part of a feasible solution of the original problem U-BRP, a constraint of the form

$$\sum_{(i,j,k,l,n)\in X^*_c}\bar{x}_{ijklnc} + \sum_{(i,j,n)\in Y^*_c}\bar{y}_{ijnc} \leq \gamma_c - 1 + \sum_{(i,j,k,l,n)\in X\setminus X^*_c}\bar{x}_{ijklnc} + \sum_{(i,j,n)\in Y\setminus B^*_{(c-1)}}\bar{b}_{ijn(c-1)} + \sum_{(i,j,n)\in Y\setminus B^*_c}\bar{b}_{ijnc} \tag{15}$$

is added to the U-BRP$_{\text{master}}$. By applying a constraint of the form (15), the current combination of moves (left–hand side) within a transition is cut off from the solution space, unless (i) the combination of moves is extended by at least one further relocation ($\bar{x}$–variables on the right–hand side), or (ii) the item locations within the related configurations $c-1$ and c are not identical to the corresponding item locations of the current optimal solution ($\bar{b}$–variables on the right–hand side). In such cases, at least one variable on the right–hand side of constraint (15) takes a value greater or equal to 1, so that the respective constraint is disabled. This ensures that a combination of moves is not falsely identified to be infeasible and therefore cut off. For example, a specific combination of moves which is infeasible within a c-th transition with the two related configurations $c-1$ and c might be feasible within the same c-th transition if the item locations within the two related configurations $c-1$ and c are changed.

In our example (Fig. 3), there are two moves within the first transition and therefore γ_1 takes value 2. To prohibit the combination of moves within the first transition the additional constraint

$$\begin{aligned}\bar{x}_{231251} + \bar{y}_{1211} \leq 1 + &\sum_{(i,j,k,l,n)\in X\setminus\{(2,3,1,2,5)\}} \bar{x}_{ijkln1} \\ &+ \sum_{\substack{(i,j,n) \in Y \setminus \{(1,1,3),(1,2,1),(1,3,4),\\ (2,1,2),(2,2,6),(2,3,5)\}}} \bar{b}_{ijn0} \\ &+ \sum_{\substack{(i,j,n) \in Y \setminus \{(1,1,3),(1,2,5),(1,3,4),\\ (2,1,2),(2,2,6)\}}} \bar{b}_{ijn1}\end{aligned}$$

is added to the U-BRP$_{\text{master}}$. Thus, the current combination of moves ($\bar{x}^*_{231251} = 1$ and $\bar{y}^*_{1211} = 1$) can only remain if at least one further relocation (different from $\bar{x}_{231251}$) is added to the first transition or if the item locations within the configuration $c-1$ (or c) differ. Note that $\bar{b}_{ijn0}$ are parameters and the corresponding term always has the value 0. Thus, it could actually be omitted from constraint (15) for $c=1$.

There also exists no feasible order for the moves of the second transition and therefore additionally the constraint

$$\begin{aligned}\bar{x}_{112132} + \bar{x}_{221162} + \bar{y}_{2122} \leq 2 + &\sum_{\substack{(i,j,k,l,n) \in X \setminus \{(1,1,2,1,3),\\ (2,2,1,1,6)\}}} \bar{x}_{ijkln2} \\ &+ \sum_{\substack{(i,j,n) \in Y \setminus \{(1,1,3),(1,2,5),(1,3,4),\\ (2,1,2),(2,2,6)\}}} \bar{b}_{ijn1} \\ &+ \sum_{\substack{(i,j,n) \in Y \setminus \{(1,1,6),(1,2,5),\\ (1,3,4),(2,1,3)\}}} \bar{b}_{ijn2}\end{aligned}$$

is added to the U-BRP$_{\text{master}}$. The third transition does not need to be considered since it includes only one move. By generating and adding these constraints to the U-BRP$_{\text{master}}$, the first iteration is finished. Iteration 2 begins with optimally solving the extended U-BRP$_{\text{master}}$. The optimal solution of the U-BRP$_{\text{master}}$ calculated in iteration 2 is presented in the upper half of Fig. 4. The first and third transitions again contain multiple moves and therefore lead to further subproblems which must be solved.

For clarity reasons, let the following three points recap and define one iteration of the procedure: (1) Optimally solve the U-BRP$_{\text{master}}$. (2) For each transition of the optimal solution of the U-BRP$_{\text{master}}$ that contains multiple moves, solve a subproblem. (3) For each subproblem that detects infeasibility, generate and add a constraint to the U-BRP$_{\text{master}}$.

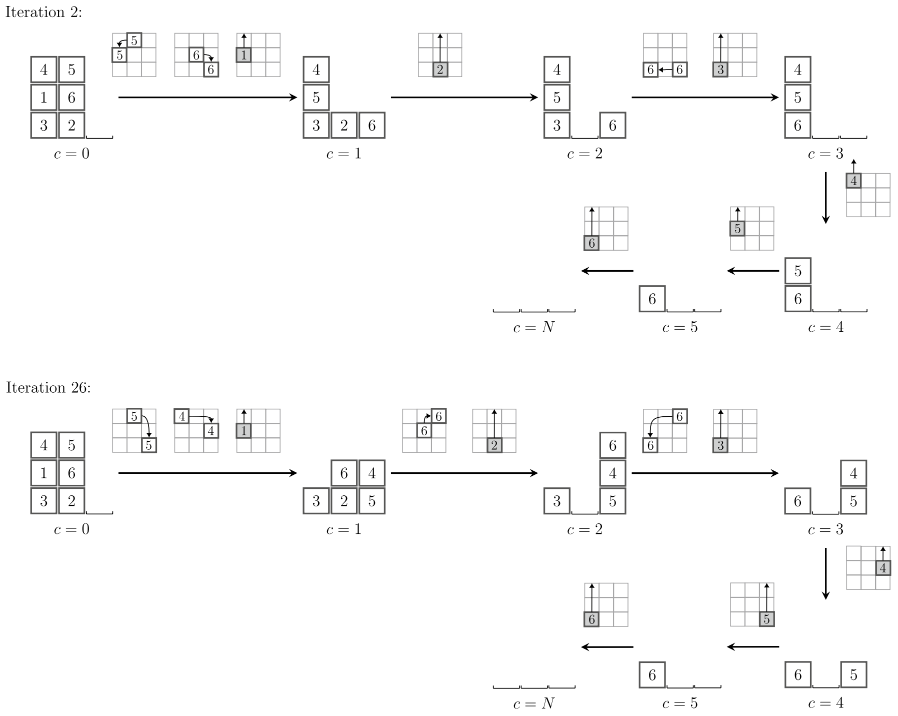

Fig. 4 Optimal solutions provided by the U-BRP$_{master}$ embedded within the row generation framework in iteration 2 and 26, respectively. The bottom solution is optimal with respect to the original problem

The procedure is repeated until the optimal solution provided by the U-BRP$_{master}$ is also feasible for the original problem U-BRP, i.e., no subproblem detects infeasibility. This basic version of the procedure needs 26 iterations (when implemented in AMPL/Gurobi) to find the solution which is presented in the bottom half of Fig. 4. Note again that this Sect. 4.1 shall explain the general outline of the procedure. Any existing infeasibility within the transitions of the example is identified by just looking closely, so that the constraints are generated without the help of a model formulation. Indeed, the detection of infeasibility and the corresponding separation of constraints is done by means of a model formulation which is presented in the next Sect. 4.2.

4.2 Separation problem

A separation of constraints of the form (15) is performed by means of a model formulation called U-BRP$_{sub(\bar{c})}$. The master problem U-BRP$_{master}$ and the subproblem U-BRP$_{sub(\bar{c})}$ both include configurations and transitions. For clarity reasons and from now on, a configuration (transition) of the U-BRP$_{master}$ is called *m-configuration* (*m-transition*) and a configuration (transition) of the U-BRP$_{sub(\bar{c})}$ is called *s-configuration* (*s-transition*). In contrast to the U-BRP$_{\text{master}}$ and its m-configurations, the U-BRP$_{\text{sub}(\bar{c})}$ contains an s-configuration after each single move to easily verify the feasibility of all moves. The U-BRP$_{\text{sub}(\bar{c})}$ is independently solved for each m-transition $1, \ldots, N-W$ of the U-BRP$_{\text{master}}$ whereby the U-BRP$_{\text{sub}(\bar{c})}$ corresponds to the $\bar{c}$-th m-transition of the U-BRP$_{\text{master}}$. Note that the notation $\bar{c}$ only appears in connection with the U-BRP$_{\text{sub}(\bar{c})}$ and allows to distinguish between the s-transitions of the U-BRP$_{\text{sub}(\bar{c})}$ and the corresponding m-transition $\bar{c}$ of the U-BRP$_{\text{master}}$.

Given an optimal solution of the U-BRP$_{\text{master}}$ with $N-W$ m-transitions and its contained moves. The aim of the subproblem U-BRP$_{\text{sub}(\bar{c})}$, $\bar{c}=1,\ldots,N-W$ is to find a sequence of the moves contained in the $\bar{c}$-th m-transition which leads to a feasible s-configuration after each single move. An obvious approach might be to add the following constraints (16) to the U-BRP$_{\text{sub}(\bar{c})}$ which allow at most one move per s-transition:

$$\sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{k=1}^{W}\sum_{l=1}^{H}\sum_{n=1}^{N} x_{ijklnc} + \sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{n=1}^{N} y_{ijnc} \leq 1 \qquad c = 1,\ldots,\gamma_{\bar{c}} \tag{16}$$

Nevertheless, this straightforward formulation (16) would be insufficient. The available moves for the U-BRP$_{\text{sub}(\bar{c})}$ are limited to the moves which are used within the corresponding $\bar{c}$-th m-transition of the U-BRP$_{\text{master}}$ and therefore the constraints (16) might cause subproblems which have no feasible solution. To guarantee feasibility of each subproblem U-BRP$_{\text{sub}(\bar{c})}$16', the constraints (16) are reformulated as soft constraints () by introducing further variables $\lambda_c \geq 0, c = 1,\ldots,\gamma_{\bar{c}}$. Within an optimal solution of the U-BRP$_{\text{sub}(\bar{c})}$, the variables λ_c indicate if the moves within the $\bar{c}$ -th m-transition are feasible with respect to the original problem U-BRP (all $\lambda_c = 0$) or not (at least one $\lambda_c > 0$).

The number of s-configurations which must be considered within the U-BRP$_{\text{sub}(\bar{c})}$ (additionally to the initial s-configuration) is given by the number of moves $\gamma_{\bar{c}}$ within the $\bar{c}$-th m-transition, calculated as presented in (14). Furthermore, the initial s-configuration $c = 0$ and the final s-configuration $c = \gamma_{\bar{c}}$ of the U-BRP$_{\text{sub}(\bar{c})}$ are determined by the m-configurations $\bar{c}-1$ and $\bar{c}$ of the U-BRP$_{\text{master}}$, respectively.

The notation used within the U-BRP$_{\text{sub}(\bar{c})}$ is like in the U-BRP$_{\text{master}}$. For clarity reasons, there is a small adjustment to clearly mark notations related to the U-BRP$_{\text{sub}(\bar{c})}$ or the U-BRP$_{\text{master}}$, e.g., x_{ijklnc} is used within the U-BRP$_{\text{sub}(\bar{c})}$ and $\bar{x}_{ijklnc}$ within the U-BRP$_{\text{master}}$. Finally, the subproblem U-BRP$_{\text{sub}(\bar{c})}$, $\bar{c}=1,\ldots,N-W$ can be stated as follows:

$$\text{U-BRP}_{\text{sub}(\bar{c})}: \quad \max \sum_{c=1}^{\gamma_{\bar{c}}} (v_{\bar{c}c} - \gamma_{\bar{c}}\lambda_c) \tag{17}$$

s.t.

$$\sum_{i=1}^{W}\sum_{j=1}^{H} b_{ijnc} + v_{nc} = 1 \qquad n = 1,\ldots,N; c = 1,\ldots,\gamma_{\bar{c}} \tag{18}$$

$$\sum_{n=1}^{N} b_{ijnc} \leq 1 \qquad i = 1, \dots, W; j = 1, \dots, H; c = 1, \dots, \gamma_{\bar{c}} - 1 \tag{19}$$

$$\sum_{n=1}^{N} b_{ijnc} \geq \sum_{n=1}^{N} b_{i(j+1)nc} \qquad i = 1, \dots, W; j = 1, \dots, H - 1; \quad c = 1, \dots, \gamma_{\bar{c}} - 1 \tag{20}$$

$$\sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{W} \sum_{l=1}^{H} \sum_{n=1}^{N} x_{ijklnc} + \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{n=1}^{N} y_{ijnc} \leq 1 + \lambda_c \qquad c = 1, \dots, \gamma_{\bar{c}} \tag{16'}$$

$$b_{ijnc} = b_{ijn(c-1)} + \sum_{k=1}^{W} \sum_{l=1}^{H} x_{klijnc} - \sum_{k=1}^{W} \sum_{l=1}^{H} x_{ijklnc} - y_{ijnc} \quad i = 1, \dots, W; j = 1, \dots, H; n = 1, \dots, N; c = 1, \dots, \gamma_{\bar{c}} \tag{21}$$

$$v_{\bar{c}c} = \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{c'=1}^{c} y_{ij\bar{c}c'} \qquad c = 1, \dots, \gamma_{\bar{c}} \tag{22}$$

$$\sum_{c=1}^{\gamma_{\bar{c}}} x_{ijklnc} = \bar{x}^*_{ijkln\bar{c}} \qquad i, k = 1, \dots, W; j, l = 1, \dots, H; \quad c = 1, \dots, N : \bar{x}^*_{ijkln\bar{c}} \geq 1 \tag{23}$$

$$x_{ijklnc} = 0 \qquad i, k = 1, \dots, W; j, l = 1, \dots, H; n = 1, \dots, N; \quad c = 1, \dots, \gamma_{\bar{c}} : \bar{x}^*_{ijkln\bar{c}} = 0 \tag{24}$$

$$y_{ijnc} = 0 \qquad i = 1, \dots, W; j = 1, \dots, H; n = 1, \dots, N; \quad c = 1, \dots, \gamma_{\bar{c}} : \bar{y}^*_{ijn\bar{c}} = 0 \tag{25}$$

$$b_{ijnc} \geq 0 \qquad i = 1, \dots, W; j = 1, \dots, H; n = 1, \dots, N; c = 1, \dots, \gamma_{\bar{c}} - 1 \tag{26}$$

$$v_{\bar{c}c} \geq 0 \qquad c = 1, \dots, \gamma_{\bar{c}} \tag{27}$$

$$x_{ijklnc} \in \{0, 1\} \qquad i, k = 1, \dots, W; j, l = 1, \dots, H; n = 1, \dots, N; \quad c = 1, \dots, \gamma_{\bar{c}} \tag{28}$$

$$y_{ijnc} \in \{0, 1\} \qquad i = 1, \dots, W; j = 1, \dots, H; n = 1, \dots, N; c = 1, \dots, \gamma_{\bar{c}} \tag{29}$$

$$\lambda_c \geq 0 \qquad c = 1, \dots, \gamma_{\bar{c}} \tag{30}$$

The constraints (18)–(22) and (24) are fundamental restrictions of the U-BRP which are already known from the U-BRP$_{\text{master}}$, constraints (3)–(8).

Each subproblem U-BRP$_{\text{sub}(\bar{c})}$ corresponds to the $\bar{c}$-th m-transition of an optimal solution of the U-BRP$_{\text{master}}$ which may contain several relocations and always contains exactly one retrieval, namely the retrieval of item $n = \bar{c}$. The number of moves $\gamma_{\bar{c}}$ within the $\bar{c}$-th m-transition determines the number of s-transitions or, respectively, the number of s-configurations which are considered within the U-BRP$_{\text{sub}(\bar{c})}$ (additionally to the initial s-configuration $c = 0$). Thus, most decision variables and constraints are defined for $c = 1, \dots, \gamma_{\bar{c}}$.

The objective (17) maximizes the number of s-configurations in which the target item $\bar{c}$ is no longer located within the stacking area. This expression is equivalent to a minimization of the number of relocations. As already mentioned, only the retrieval of item $n = \bar{c}$ takes place within the $\bar{c}$-th m-transition and therefore within the U-BRP$_{\text{sub}(\bar{c})}$, but it is unknown in which s-transition the retrieval takes place. Thus, v_{nc}, $n = 1, \dots, N$, $c = 1, \dots, \gamma_{\bar{c}}$ is a variable iff $n = \bar{c}$ and so, only $v_{\bar{c}c}$ occurs in the objective (17). All items $n < \bar{c}$ are already retrieved in preceding m-transitions and all items $n > \bar{c}$ will be retrieved in later m-transitions whereby all v_{nc} with $n \neq \bar{c}$ are parameters defined as follows:

$$v_{nc} := \begin{cases} 0, & n = \bar{c}+1, \dots, N; c = 1, \dots, \gamma_{\bar{c}} \\ 1, & n = 1, \dots, \bar{c}-1; c = 1, \dots, \gamma_{\bar{c}} \end{cases} \qquad \bar{c} = 1, \dots, N - W \tag{31}$$

The second part of the objective (17) is a penalization term which is associated with the constraints (16’). If the c-th s-transition of the U-BRP$_{\text{sub}(\bar{c})}$ contains more than one move, then the corresponding variable λ_c takes a value greater than 0 which, in turn, is penalized in the objective (17). Note that the variables λ_c are always integer since their values are determined by binary variables only. The weights within the penalty term ($\gamma_{\bar{c}}\lambda_c$) are set to the number of moves $\gamma_{\bar{c}}$. These are the minimum values required to ensure that an optimal solution of the U-BRP$_{\text{sub}(\bar{c})}$ violates one or more constraints of (16) only if there exists no feasible solution which does not violate (16), i.e., if there is no feasible solution with $\lambda_c = 0$ for all $c = 1, \dots, \gamma_{\bar{c}}$ (see the Appendix 2 for a proof).

Furthermore, the parameters b_{ijn0} and $b_{ijn\gamma_{\bar{c}}}$ which determine the initial and final s-configuration of the U-BRP$_{\text{sub}(\bar{c})}$ are defined as follows:

$$b_{ijn0} := \bar{b}^{*}_{ijn(\bar{c}-1)} \qquad i = 1, \dots, W; j = 1, \dots, H; n = 1, \dots, N \tag{32}$$

$$b_{ijn\gamma_{\bar{c}}} := \bar{b}^{*}_{ijn\bar{c}} \qquad i = 1, \dots, W; j = 1, \dots, H; n = 1, \dots, N \tag{33}$$

Thus, the origin m-configuration of the $\bar{c}$-th m-transition (m-configuration $\bar{c} - 1$) defines the initial s-configuration $c = 0$ of the subproblem U-BRP$_{\text{sub}(\bar{c})}$ (by (32)) and analogously the resulting m-configuration of the $\bar{c}$-th m-transition (m-configuration $\bar{c}$) defines the final s-configuration $c = \gamma_{\bar{c}}$ of the subproblem U-BRP$_{\text{sub}(\bar{c})}$ (by (33)).

Finally, the constraints (24) and (25) enforce that only those moves can be used within the U-BRP$_{\text{sub}(\bar{c})}$ which are also used within the corresponding m-transition. Thereby, the majority of the variables x_{ijklnc} and y_{ijnc} is fixed to the value 0. Additionally, the constraints (23) ensure that each relocation of the $\bar{c}$-th m-transition takes place within the U-BRP$_{\text{sub}(\bar{c})}$. In most cases the constraints (23) are superfluous since the available relocations (those which are not prohibited by (24)) have to be used anyway to reach the given final s-configuration, but in a few cases the final s-configuration within a subproblem can be obtained by using only a subset of these relocations. In this scenario, a relaxation of constraints (23) may result in a generation of identical constraints within consecutive iterations and thus the procedure does not terminate. In turn, integrating the constraints (23) into the U-BRP$_{\text{sub}(\bar{c})}$ guarantees that the procedure can find an optimal solution within a finite number of iterations (see the Appendix 3 for an example in which the constraints (23) are mandatory as well as for a clarification that the procedure never causes deadlocks if the constraints (23) are enabled). Similar constraints for the retrievals are not needed. Within each subproblem the given final s-configuration always includes one item less than the given initial s-configuration and there is always exactly one retrieval permitted. Thus, the final s-configuration cannot be obtained if the respective retrieval is not a part of the solution.

The example in Fig. 5 gives an overview of the connection between the master problem and the subproblems. The upper row shows an optimal solution of the

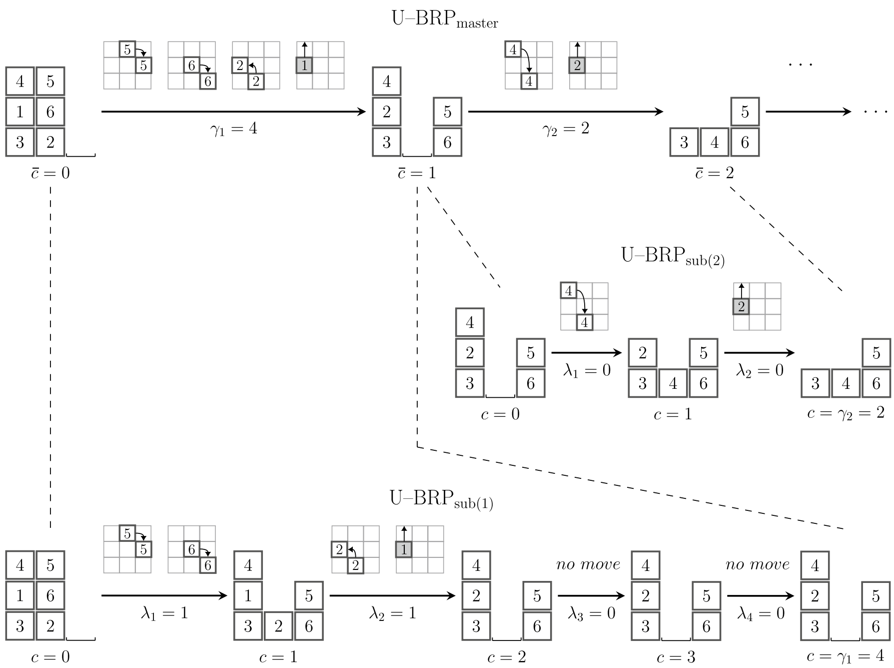

Fig. 5 First two m-transitions of an optimal solution of the U-BRP$_{\text{master}}$ and optimal solutions of the two corresponding subproblems U-BRP$_{\text{sub}(1)}$ and U-BRP$_{\text{sub}(2)}$

U-BRP$_{\text{master}}$ which occurs within the procedure. Only the first two m-transitions contain relocations and are therefore presented. Furthermore, the corresponding subproblems U-BRP$_{\text{sub}(1)}$ and U-BRP$_{\text{sub}(2)}$ are presented. Since the first m-transition contains $\gamma_1 = 4$ moves, the corresponding subproblem U-BRP$_{\text{sub}(1)}$ considers the s-configurations $0, \dots, 4$ and therefore four s-transitions. Analogously, the U-BRP$_{\text{sub}(2)}$ considers the s-configurations $0, \dots, 2$ and therefore two s-transitions. The optimal solution of the first subproblem contains variables $\lambda_c > 0$ so there is no feasible solution which leads to a feasible s-configuration after each single move. Thus, the combination of moves cannot be part of a feasible solution of the original problem U-BRP and therefore is cut off by adding the constraint

$$\begin{aligned}
\bar{x}_{233251} + \bar{x}_{223161} + \bar{x}_{211221} + \bar{y}_{1211} \leq 3 + & \sum_{\substack{(i,j,k,l,n) \in X \setminus \{(2,3,3,2,5),\\ (2,2,3,1,6),\\ (2,1,1,2,2)\}}} \bar{x}_{ijkln1} \\
+ & \sum_{\substack{(i,j,n) \in Y \setminus \{(1,1,3),(1,2,1),\\ (1,3,4),(2,1,2),\\ (2,2,6),(2,3,5)\}}} \bar{b}_{ijn0} \\
+ & \sum_{\substack{(i,j,n) \in Y \setminus \{(1,1,3),(1,2,2),\\ (1,3,4),(3,1,6),\\ (3,2,5)\}}} \bar{b}_{ijn1}
\end{aligned}$$

to the U-BRP$_{\text{master}}$. On the other hand, the optimal solution of the second subproblem contains $\lambda_c = 0$ for all $c = 1, \dots, \gamma_2$. Thus, the combination of moves of the second m-transition is identified to be feasible with respect to the original problem and therefore no constraint is generated.

5 Acceleration methods

The row generation framework presented in Sect. 4 is a basic version. In this section, we propose possible modifications which may accelerate the procedure. In Sect. 5.1, we present possibilities to fix a part of the variables within the U-BRP$_{\text{master}}$ to the value 0 in a preprocessing step. In the Sect. 5.2, we present valid inequalities for the U-BRP$_{\text{master}}$. In the Sects. 5.3 and 5.4, variable fixings and symmetry breaking constraints for the U-BRP$_{\text{sub}(\bar{c})}$ are presented. Finally, in Sect. 5.5, we propose three different types of constraints which can be generated within the procedure.

5.1 Fixing the variables of the master problem

Some moves can be excluded in advance and therefore the corresponding variables of the U-BRP$_{\text{master}}$ can be fixed to the value 0 in a preprocessing step. These variables can actually be omitted from the model, but for a better readability we leave the U-BRP$_{\text{master}}$ unchanged and state this by the definitions (34)–(36).

The first fixings (34) concern the variables $\bar{b}_{ijnc}$.

$$\bar{b}_{ijnc} := 0 \qquad i = 1, \dots, W; j = 1, \dots, H; n, c = 1, \dots, N - W : \\ n \le c \ \vee \ j > N - c \tag{34}$$

Remember that item c is the target item within the c-th m-transition. Thus, all items $n \le c$ are not anymore located within m-configuration c. Second, an item n cannot be located on a tier j in an m-configuration c if there are not enough items left within that m-configuration to reach that tier. Therefore, in an m-configuration c there is never an item n located on a tier j higher than the number $N - c$ of remaining items in that m-configuration.

The variables $\bar{x}_{ijklnt}$ are fixed as denoted by (35). There are four cases in which a variable is fixed.

$$\bar{x}_{ijklnc} := 0 \qquad i, k = 1, \dots, W; j, l = 1, \dots, H; n, c = 1, \dots, N - W : \\ i = k \ \vee \ n \le c \ \vee \ j + l = 2 \ \vee \ j + l > N - c + 2 \tag{35}$$

The first condition $i = k$ of (35) prohibits a relocation within one stack since such a relocation is never feasible. Note that by applying these fixings, the constraints (8) become redundant and can therefore be relaxed. Second, within the c-th m-transition the target item c is never relocated but only retrieved since a relocation of this item is superfluous. Additionally, all items $n < c$ are already retrieved and therefore cannot be relocated anymore. Thus, all relocations with $n \le c$ can be fixed to 0. Third, a relocation of an item from tier $j = 1$ of one stack to tier $l = 1$ of another stack is superfluous since it is indifferent in which stack an item is located if it is on the bottom. Fourth, a relocation is excluded from the solution space if the sum $j + l$ of the corresponding tiers of this relocation cannot be reached by the remaining items. Remember that the variables $\bar{x}_{ijklnc}$ represent relocations within the c-th m-transition originating in m-configuration $c - 1$. Thus, the number of the remaining items within the concerned m-configuration is $N - c + 1$. The sum $j + l$ connected to a relocation can have a maximal value $N - c + 2$ (number of the remaining items plus 1) which occurs in an m-configuration with all items located in at most two stacks. Figure 6 gives two examples to illustrate the fourth condition. The items are not numbered since it is irrelevant for illustration purposes. Both examples include $N = 3$ items and concern relocations within the first m-transition ($c = 1$). Thus, $j + l$ can have a

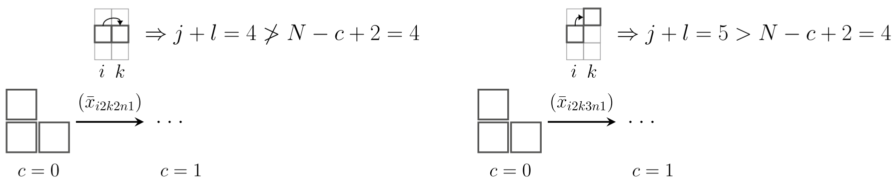

Fig. 6 The relocation on the left is not affected by a variable fixing. The relocation on the right is prohibited by fixing the corresponding variable to the value 0

maximal value of $N-c+2=4$. The example on the left shows a relocation which is not affected by a fixing. On the right side an example for a relocation is presented which is prohibited by fixing the corresponding variable to the value 0.

The variables $\bar{y}_{ijnc}$ are fixed as stated by (36).

$$\bar{y}_{ijnc} := 0 \qquad i=1,\dots,W; j=1,\dots,H; n,c=1,\dots,N-W : \\ n \neq c \quad \vee \quad j > N-c+1 \tag{36}$$

Within the c-th m-transition only item $n=c$ is retrieved and thus the variables $\bar{y}_{ijnc}$ are fixed to the value 0 if $n \neq c$. Since the variables $\bar{y}_{ijnc}$ remain unfixed only if $n=c$, they can be reduced to triple indexed variables $\bar{y}_{ijc}$, but for clarity reasons we leave the variables unchanged for the remainder of this paper. Second, an item cannot be retrieved from a location on a tier j within the c-th m-transition if the number $N-c+1$ of items left within m-configuration $c-1$ is too low to reach that tier.

5.2 Valid inequalities for the master problem

Furthermore, we formulate valid inequalities for strengthening the relaxed model formulation U-BRP$_{\text{master}}$. The intention is to cut off solutions of the solution space which are infeasible with respect to the original problem U-BRP. Since these solutions may be prohibited within the row generation procedure otherwise, it should reduce the number of iterations required within the procedure.

Property 5.1 *If there is a relocation of an item n in an m-transition c ending in slot (i,j), then slot (i,j) must not be occupied by another item $n' \neq n$ or there must be a further relocation of an item $n' \neq n$ in that m-transition c starting in slot (i,j) or there must be a retrieval of item c from slot (i,j).*

With respect to the U-BRP$_{\text{master}}$ it is feasible to relocate an item n to a *slot* (i,j) which is actually occupied by another item $n' \neq n$ if item n leaves the slot again by a further relocation within the same m-transition. For the original problem U-BRP, this would cause an infeasible solution since it violates the condition that a slot must be occupied by at most one item. Therefore, we are applying the valid inequalities (37) to implement the insight of Property 5.1.

$$\sum_{k=1}^{W}\sum_{l=1}^{H} \bar{x}_{klijnc} \leq M\left(\left(1 - \sum_{\substack{n'=1\\ n'\neq n}}^{N} \bar{b}_{ijn'(c-1)}\right) + \sum_{k=1}^{W}\sum_{l=1}^{H}\sum_{\substack{n'=1\\ n'\neq n}}^{N} \bar{x}_{ijkln'c} + \bar{y}_{ijcc}\right) \tag{37}$$
$$i=1,\dots,W; j=1,\dots,H; n=1,\dots,N; c=1,\dots,N-W$$

Note that the variables $\bar{b}_{ijn(c-1)}$ are excluded from the constraints (37) to not prohibit some sort of "back and forth"- or cyclic relocations within an m-transition.

Relocations are called *cyclic relocations* if an item is relocated at least twice within an m-transition c whereby it is relocated to the slot in which it was initially located in the originating m-configuration $c-1$. For specific instances, cyclic relocations are mandatory to obtain an optimal solution and therefore shall not be cut off (see the Appendix 4 for an example).

Property 5.2 *If there is a relocation in an m-transition c starting or ending in slot* (i, j), *then the slot* $(i, j+1)$ *above must either be a free slot or there must be another relocation in that m-transition c starting in slot* $(i, j+1)$ *or there must be a retrieval of item c from slot* $(i, j+1)$. *Analogously, if item c is retrieved from slot* (i, j) *in m-transition c, then the slot* $(i, j+1)$ *above must either be a free slot or there must be another relocation in that m-transition c starting in slot* $(i, j+1)$.

On the basis of Property 5.2 we are applying the valid inequalities (38) and (39).

$$\begin{aligned}
&\sum_{k=1}^{W}\sum_{l=1}^{H}\sum_{n=1}^{N}(\bar{x}_{ijklnc}+\bar{x}_{klijnc}) \\
&\quad \leq M\left(\left(1-\sum_{n=1}^{N}\bar{b}_{i(j+1)n(c-1)}\right)\right. \\
&\qquad \left. +\sum_{k=1}^{W}\sum_{l=1}^{H}\sum_{n=1}^{N}\bar{x}_{i(j+1)klnc}+\bar{y}_{i(j+1)cc}\right) \\
&\qquad i=1,\dots,W; j=1,\dots,H-1; c=1,\dots,N-W
\end{aligned} \tag{38}$$

$$\begin{aligned}
&\bar{y}_{ijcc} \leq \left(1-\sum_{n=1}^{N}\bar{b}_{i(j+1)n(c-1)}\right)+\sum_{k=1}^{W}\sum_{l=1}^{H}\sum_{n=1}^{N}\bar{x}_{i(j+1)klnc} \\
&\qquad i=1,\dots,W; j=1,\dots,H-1; c=1,\dots,N-W
\end{aligned} \tag{39}$$

Note that a Big–M is not needed in the constraints (39) since only one retrieval takes place within each m-transition and therefore the left–hand sides have a maximal value 1.

Property 5.3 *For all* $c=1,\dots,N-W$ *it can be enforced that the retrieval of the target item c is the last move within the c-th m-transition, i.e., if the target item c is located in slot* (i, j) *at the beginning of the c-th m-transition, then this slot* (i, j) *and all slots below can be disabled for any relocation in this m-transition. Such relocations can take place in the following m-transition and therefore no feasible solution of the original problem gets lost.*

Based on the Property 5.3 we are applying the valid inequalities (40) to enforce that in each m-transition $c=1,\dots,N-W$ there are neither relocations starting or ending in the slot in which the target item c is located nor in a slot below.

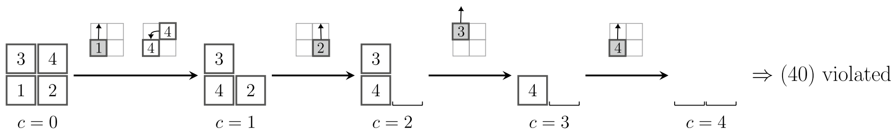

Fig. 7 Infeasible solution with respect to the original problem U-BRP

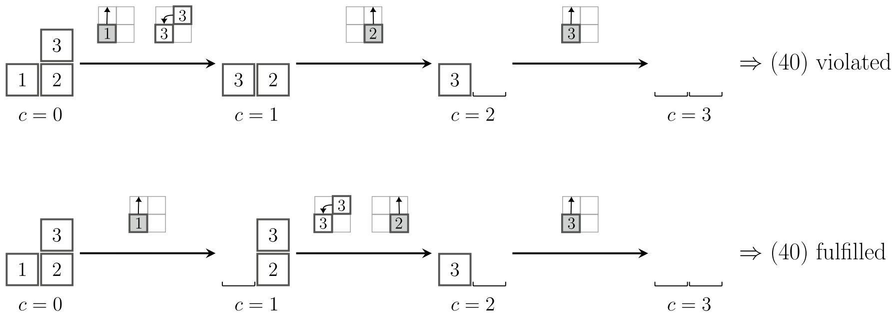

Fig. 8 Symmetrical solutions of the U-BRP$_{\text{master}}$

$$\sum_{j'=1}^{j}\sum_{k=1}^{W}\sum_{l=1}^{H}\sum_{n=1}^{N}(\bar{x}_{ij'klnc}+\bar{x}_{klij'nc}) \leq M(1-\bar{b}_{ijc(c-1)})$$
$$i=1,\dots,W; j=1,\dots,H; c=1,\dots,N-W \tag{40}$$

The valid inequalities (40) have two effects. A first (obvious) effect is that some solutions are cut off from the solution space which are infeasible with respect to the original problem U-BRP. An example for such a solution is presented in Fig. 7.

A second effect is that feasible but symmetrical solutions may be prevented. A minimal example for symmetrical solutions is depicted in Fig. 8. Both solutions are feasible with respect to the U-BRP but the upper solution is prohibited by (40).

Taking together the Properties 5.1 – 5.3, the valid inequalities (37), (38) and (40) can be reduced to (37'), (38') and (40'), respectively, without loss in strength of the valid inequalities.

$$\sum_{k=1}^{W}\sum_{l=1}^{H}\bar{x}_{klijnc} \leq M\left(\left(1-\sum_{\substack{n'=1\\ n'\neq n}}^{N}\bar{b}_{ijn'(c-1)}\right)+\sum_{k=1}^{W}\sum_{l=1}^{H}\sum_{\substack{n'=1\\ n'\neq n}}^{N}\bar{x}_{ijkln'c}\right) \tag{37'}$$
$$i=1,\dots,W; j=1,\dots,H; n=1,\dots,N; c=1,\dots,N-W$$

$$\sum_{k=1}^{W}\sum_{l=1}^{H}\sum_{n=1}^{N}\bar{x}_{ijklnc} \leq M\left(\left(1-\sum_{n=1}^{N}\bar{b}_{i(j+1)n(c-1)}\right)+\sum_{k=1}^{W}\sum_{l=1}^{H}\sum_{n=1}^{N}\bar{x}_{i(j+1)klnc}\right) \quad i=1,\dots,W; j=1,\dots,H-1; c=1,\dots,N-W \tag{38'}$$

$$\sum_{k=1}^{W}\sum_{l=1}^{H}\sum_{n=1}^{N}\bar{x}_{ijklnc} \leq M(1-\bar{b}_{ijc(c-1)}) \quad i=1,\dots,W; j=1,\dots,H; c=1,\dots,N-W \tag{40'}$$

For clarity reasons, this reduction is explained in three steps. (1) According to Property 5.3, within an m-transition, relocations shall neither start or end in the slot of the target item c nor in a slot below, i.e., a retrieval from a *slot* (i, j) shall neither enable this *slot* (i, j) nor a slot below for any relocation. Therefore, the $\bar{y}$–variables can be omitted from the constraints (37) and (38). (2) Furthermore, the valid inequalities (40) are reduced by just disabling the slot, in which the target item is located, for any relocation. All slots below are implicitly disabled by the constraints (38'). (3) Finally, the valid inequalities (38) and (40) may restrict any relocation which starts or ends in a specific *slot* (i, j). These constraints can be reduced by only prohibiting any relocation which starts in that specific *slot* (i, j) (and not those which end there) and therefore the variables $\bar{x}_{klijnc}$ and $\bar{x}_{klij'nc}$ are omitted from the left–hand sides of constraints (38) and (40), respectively. A relocation ending in an occupied *slot* (i, j) automatically leads to a further relocation starting in that *slot* (i, j) due to constraints (37'). Therefore, if a relocation starting in a *slot* (i, j) is prohibited by (38') or (40'), then all relocations ending in this *slot* (i, j) are implicitly prohibited by the constraints (37'). Note that the reduced constraints (37'), (38') and (40') are mutually dependent and therefore should be applied together and not isolated.

The constraints (41) affect the retrieval moves. Within the c-th m-transition, as soon as the target item c is accessible it can be retrieved and therefore a relocation of it is always superfluous within that m-transition. Therefore, the constraints (41) enforce that item c is retrieved from the *slot* (i, j) where it is located at the beginning of the c-th m-transition.

$$\bar{y}_{ijcc} = \bar{b}_{ijc(c-1)} \qquad i=1,\dots,W; j=1,\dots,H; c=1,\dots,N-W \tag{41}$$

Note that a target item is never relocated within an optimal solution of the original problem U-BRP or the U-BRP$_{\text{master}}$ (in its initial version) since such relocations are superfluous. Within the row generation procedure such relocations might occur. The reason is that a superfluous relocation may (re-)allow a combination of moves which is actually prohibited by a generated constraint (see the relocations of item 3 in the Appendix 3 for an illustration of a similar effect). Additionally, these superfluous relocations are implicitly prohibited by the valid inequalities (40) or (40') which actually make the constraints (41) redundant. Nevertheless, applying (41) might reduce solution time.

5.3 Fixing the variables of the subproblems

As in the master problem U-BRP$_{\text{master}}$, there are some variables within the subproblems U-BRP$_{\text{sub}(\bar{c})}$ which can be fixed in a preprocessing step. Note that fixing the variables of the U-BRP$_{\text{master}}$ as presented in Sect. 5.1 implicitly fixes the corresponding variables of the U-BRP$_{\text{sub}(\bar{c})}$ within the row generation procedure due to the constraints (24) and (25). Nevertheless, further variables can be fixed within the U-BRP$_{\text{sub}(\bar{c})}$ which may not yet be affected.

Due to the Property 5.3, the retrieval of the target item can be assumed to be the last move within the respective m-transition. Thus, for each U-BRP$_{\text{sub}(\bar{c})}$, $\bar{c} = 1, \dots, N - W$ it can be enforced by the fixings (42) that the retrieval of the target item $\bar{c}$ takes place in the last s-transition $c = \gamma_{\bar{c}}$ and from the *slot* (i, j) where the target item is initially located. In turn, all retrievals which are related to the target item $\bar{c}$ and to its initial location but not to the last s-transition $c = \gamma_{\bar{c}}$ can be prohibited by the fixings (43). All retrievals which are not related to the target item $\bar{c}$ or not to the initial location of the target item are already excluded within the row generation procedure by the constraints (25) and are therefore not part of this subsection.

$$y_{ij\bar{c}\gamma_{\bar{c}}} := 1 \qquad i = 1, \dots, W; j = 1, \dots, H \, : \, \bar{b}^*_{ij\bar{c}(\bar{c}-1)} = 1 \tag{42}$$

$$y_{ij\bar{c}c} := 0 \qquad i = 1, \dots, W; j = 1, \dots, H; c = 1, \dots, \gamma_{\bar{c}} - 1 \, : \, \bar{b}^*_{ij\bar{c}(\bar{c}-1)} = 1 \tag{43}$$

Taking together the constraints (25) and the fixings (42) and (43), all variables y_{ijnc} are fixed and therefore become parameters.

According to that, it can be assumed that the target item remains in its initial location within all s-configurations $c = 1, \dots, \gamma_{\bar{c}} - 1$ except in the last one $c = \gamma_{\bar{c}}$ since it is retrieved in the last s-transition. Therefore, further variables can be fixed according to (44) and (45).

$$b_{ij\bar{c}c} := \bar{b}^*_{ij\bar{c}(\bar{c}-1)} \qquad i = 1, \dots, W; j = 1, \dots, H; c = 1, \dots, \gamma_{\bar{c}} - 1 \tag{44}$$

$$b_{ij\bar{c}\gamma_{\bar{c}}} := 0 \qquad i = 1, \dots, W; j = 1, \dots, H \tag{45}$$

Note that one could apply additional fixings to forbid any relocation of the target item, but since it is implicitly done within the row generation procedure by the constraints (24), no further x–variables are fixed within the subproblems.

Enforcing the retrieval of the target item $\bar{c}$ within the subproblem U-BRP$_{\text{sub}(\bar{c})}$ to take place in the last s-transition $c = \gamma_{\bar{c}}$, $v_{\bar{c}c}$ are not anymore variables but parameters defined as follows:

$$v_{\bar{c}c} := \begin{cases} 0, & c = 1, \dots, \gamma_{\bar{c}} - 1 \\ 1, & c = \gamma_{\bar{c}} \end{cases} \qquad \bar{c} = 1, \dots, N - W \tag{46}$$

Therefore, by applying the variable fixings (42)–(45), $v_{\bar{c}c}$ can be omitted from the objective (17) of the U-BRP$_{\mathrm{sub}(\bar{c})}$ such that the objective reduces to $\max \sum_{c=1}^{\gamma_{\bar{c}}} -\gamma_{\bar{c}}\lambda_c$ ($\Leftrightarrow \max \sum_{c=1}^{\gamma_{\bar{c}}} -\lambda_c$). Furthermore, the constraints (18) can be relaxed since v_{nc} are not anymore variables but parameters and the correct values for the variables b_{ijnc} are implicitly ensured by the constraints (21). The constraints (22) and (25) contain parameters only and can therefore also be relaxed. Note that the valid inequalities (40) or (40') within the U-BRP$_{\mathrm{master}}$ should be activated when fixing the variables within the U-BRP$_{\mathrm{sub}(\bar{c})}$ as denoted by (42)–(45) to ensure that the Property 5.3 holds true.

5.4 Symmetry breaking constraints for the subproblems

The symmetry breaking constraints (47) within the U-BRP$_{\mathrm{sub}(\bar{c})}$ can be stated as follows:

$$\sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{k=1}^{W}\sum_{l=1}^{H}\sum_{n=1}^{N} x_{ijkln(c-1)} \leq (c-1) \cdot \sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{k=1}^{W}\sum_{l=1}^{H}\sum_{n=1}^{N} x_{ijklnc} \qquad c = 2, \dots, \gamma_{\bar{c}} - 1 \tag{47}$$

If an optimal solution of the U-BRP$_{\mathrm{sub}(\bar{c})}$ contains multiple moves within an s-transition $c = 1, \dots, \gamma_{\bar{c}}$, then there must be at least one s-transition $c' \neq c$ without any move. If one or more s-transitions without any move exist, then the constraints (47) enforce that these s-transitions must take place at the beginning. The last s-transition $c = \gamma_{\bar{c}}$ is omitted from the domain of the constraints since there is always a move, namely the retrieval of item $\bar{c}$, in this s-transition (enforced by (42)). Therefore, not to perform a relocation in s-transition $\gamma_{\bar{c}}$ must not prohibit a relocation in the preceding s-transition $\gamma_{\bar{c}} - 1$. Note that the variable fixings (42)–(45) should be applied when adding the constraints (47) to the U-BRP$_{\mathrm{sub}(\bar{c})}$ to ensure the retrieval to take place within the last s-transition $\gamma_{\bar{c}}$.

The sum on the right–hand sides of constraints (47) is multiplied by the factor $(c - 1)$. If a relocation takes place in an s-transition c, then $c - 1$ is the maximal number of relocations that can take place in the preceding s-transition $c - 1$. Therefore, the factor $(c - 1)$ is large enough to not falsely cut off an optimal solution.

Note that the computational effort for optimally solving the U-BRP$_{\mathrm{sub}(\bar{c})}$ is negligible in comparison with the U-BRP$_{\mathrm{master}}$ due to fixing the most variables. Nevertheless, applying the constraints (47) might reduce solution time, especially caused by a reduced number of iterations required.

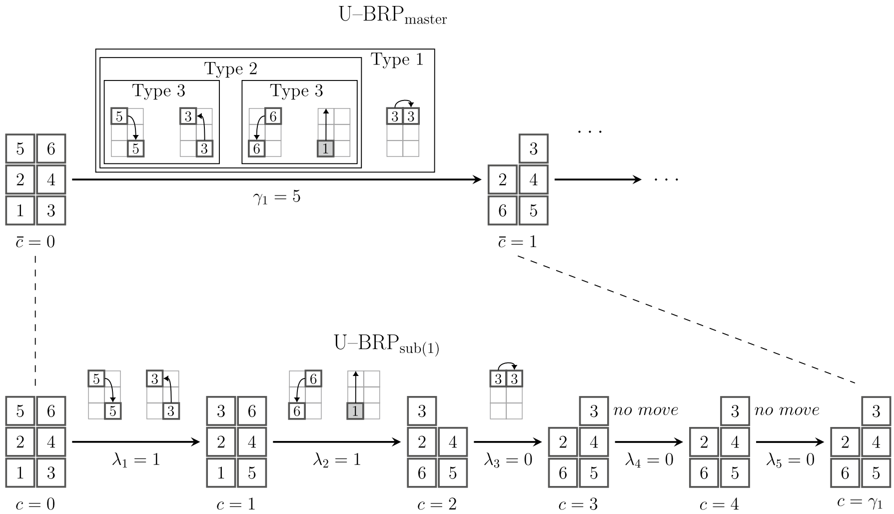

Fig. 9 First m-transition of an optimal solution of the U-BRP$_{\text{master}}$ and an optimal solution of the corresponding subproblem U-BRP$_{\text{sub}(1)}$

5.5 Alternative types of generated constraints

In this subsection, we propose three types of constraints that can be generated within the row generation procedure. Constraints of Type 1 are the generated constraints presented in Sect. 4.1. Alternatively, a constraint generation of Type 2 or 3 can be applied. For illustration purposes, Fig. 9 introduces a new example with $N = 6$ items, $W = 2$ stacks and a limited height $H = 5$. The upper row shows the first m-transition of an optimal solution for the U-BRP$_{\text{master}}$ which occurs within the row generation procedure. The bottom row shows an optimal solution of the corresponding subproblem U-BRP$_{\text{sub}(1)}$ consisting of five s-transitions. Additionally, it is illustrated which combinations of moves are prohibited by the different types of constraints. Note that the example of Fig. 9 contains the U-BRP$_{\text{master}}$ and the U-BRP$_{\text{sub}(\bar{c})}$ in its basic versions. This example cannot arise while applying the acceleration methods presented in the Sect. 5.1 – 5.4 since the solutions violate many of the valid inequalities, symmetry breaking constraints and variable fixings mentioned so far (see the Appendix 5 for further information on the fact that Type 2 is tighter than Type 1, and Type 3 is tighter than Type 2.).

Type 1 The first type of generated constraints is like the explanations and the example within Sect. 4.1. If a subproblem U-BRP$_{\text{sub}(\bar{c})}$ has an optimal solution with at least one s-transition $c = 1, \ldots, \gamma_{\bar{c}}$ containing more than one move (determined by $\lambda_c > 0$), then a constraint is added to the U-BRP$_{\text{master}}$. This constraint prohibits all moves, which are contained in the U-BRP$_{\text{sub}(\bar{c})}$, in this combination in the $\bar{c}$-th m-transition. In our example of Fig. 9, the additional constraint

$$\bar{x}_{132151} + \bar{x}_{211331} + \bar{x}_{231161} + \bar{x}_{132331} + \bar{y}_{1111} \leq 4 + \sum_{\substack{(i,j,k,l,n) \in X \setminus \{(1,3,2,1,5),(2,1,1,3,3),\\ (2,3,1,1,6),(1,3,2,3,3)\}}} \bar{x}_{ijkln1} + \sum_{\substack{(i,j,n) \in Y \setminus \{(1,1,1),(1,2,2),\\ (1,3,5),(2,1,3),\\ (2,2,4),(2,3,6)\}}} \bar{b}_{ijn0} + \sum_{\substack{(i,j,n) \in Y \setminus \{(1,1,6),(1,2,2),\\ (2,1,5),(2,2,4),\\ (2,3,3)\}}} \bar{b}_{ijn1}$$

is added to the U-BRP$_{\text{master}}$.

Type 2 The second type of generated constraints is similar to Type 1, but only such moves are considered which are contained in an s-transition $c = 1, \ldots, \gamma_{\bar{c}}$ with $\lambda_c > 0$. In this case, the additional constraint

$$\bar{x}_{132151} + \bar{x}_{211331} + \bar{x}_{231161} + \bar{y}_{1111} \leq 3 + \sum_{\substack{(i,j,k,l,n) \in X \setminus \{(1,3,2,1,5),(2,1,1,3,3),\\ (2,3,1,1,6),(1,3,2,3,3)\}}} \bar{x}_{ijkln1} + \sum_{\substack{(i,j,n) \in Y \setminus \{(1,1,1),(1,2,2),\\ (1,3,5),(2,1,3),\\ (2,2,4),(2,3,6)\}}} \bar{b}_{ijn0} + \sum_{\substack{(i,j,n) \in Y \setminus \{(1,1,6),(1,2,2),\\ (2,1,5),(2,2,4),\\ (2,3,3)\}}} \bar{b}_{ijn1}$$

is added to the U-BRP$_{\text{master}}$ of our example. Actually, the relocation $\bar{x}_{132331}$ does not cause infeasibility since it can take place as the only move within an s-transition (see the third s-transition of the U-BRP$_{\text{sub}(1)}$ in Fig. 9). Thus, the decision variable $\bar{x}_{132331}$ is not part of the left–hand side of the generated constraint, so the value of $\bar{x}_{132331}$ is not affected by this constraint. Note that constraints of Type 2 are tighter than constraints of Type 1 since they always cut off the same solutions as Type 1 constraints but prohibit further solutions if Type 1 and 2 constraints are not identical. In our example, the partial solution $\bar{x}_{132151} = 1, \bar{x}_{211331} = 1, \bar{x}_{231161} = 1, \bar{y}_{1111} = 1$ (and all other $\bar{x}_{ijkln1}$ and $\bar{y}_{ijn1}$ with value 0) is still feasible with the additional constraint of Type 1, but not anymore with the constraint of Type 2.

Type 3 Finally, in terms of Type 3, for each s-transition $c = 1, \ldots, \gamma_{\bar{c}}$ of a subproblem U-BRP$_{\text{sub}(\bar{c})}$ with $\lambda_c > 0$, an individual constraint is generated and added to the U-BRP$_{\text{master}}$. In our example (Fig. 9), there are two s-transitions with $\lambda_c > 0$ and therefore two individual constraints are generated. In the particular case, the additional constraints

$$\bar{x}_{132151} + \bar{x}_{211331} \leq 1 + \sum_{\substack{(i,j,k,l,n) \in X \setminus \{(1,3,2,1,5),(2,1,1,3,3),\\ (2,3,1,1,6),(1,3,2,3,3)\}}} \bar{x}_{ijkln1} + \sum_{\substack{(i,j,n) \in Y \setminus \{(1,1,1),(1,2,2),\\ (1,3,5),(2,1,3),\\ (2,2,4),(2,3,6)\}}} \bar{b}_{ijn0} + \sum_{\substack{(i,j,n) \in Y \setminus \{(1,1,6),(1,2,2),\\ (2,1,5),(2,2,4),\\ (2,3,3)\}}} \bar{b}_{ijn1}$$

and

$$\bar{x}_{231161} + \bar{y}_{1111} \leq 1 + \sum_{\substack{(i,j,k,l,n) \in X \setminus \{(1,3,2,1,5),(2,1,1,3,3),\\ (2,3,1,1,6),(1,3,2,3,3)\}}} \bar{x}_{ijkln1} + \sum_{\substack{(i,j,n) \in Y \setminus \{(1,1,1),(1,2,2),\\ (1,3,5),(2,1,3),\\ (2,2,4),(2,3,6)\}}} \bar{b}_{ijn0} + \sum_{\substack{(i,j,n) \in Y \setminus \{(1,1,6),(1,2,2),\\ (2,1,5),(2,2,4),\\ (2,3,3)\}}} \bar{b}_{ijn1}$$

are added to the U-BRP$_{\text{master}}$. Note that a constraint generation of Type 3 delivers the tightest constraints and therefore should lead to the lowest average number of iterations required within the row generation procedure.

5.6 Tightening the generated constraints

Remember that a generated constraint (15) does not prohibit a specific combination of moves within an m-transition if it is extended by at least one further relocation or if the adjacent m-configurations differ (see Sect. 4.1). This is ensured by the variables on the right–hand side of (15). In turn, it is very rare that any of these variables on the right–hand side take a value unequal to 0, i.e., that an optimal solution would be falsely cut off if the variables on the right–hand side of (15) were omitted. To make use of this insight, we partition the procedure in two phases and tighten the generated constraints within the first phase. Note that this modification is applicable for all three types of generated constraints presented in Sect. 5.5.

Phase 1 The procedure starts with a generation of tighter constraints denoted by (15') instead of (15).

$$\sum_{(i,j,k,l,n)\in X_c^*} \bar{x}_{ijklnc} + \sum_{(i,j,n)\in Y_c^*} \bar{y}_{ijnc} \leq \gamma_c - 1 \tag{15'}$$

With the exception of the tighter cuts (15'), the procedure is performed as presented in Sect. 4. After a termination of the procedure, the second phase starts.

Phase 2 The result of Phase 1 is the U-BRP$_{\text{master}}$ extended by all generated constraints in all iterations of Phase 1 and the corresponding optimal solution of the extended U-BRP$_{\text{master}}$. Since constraints of the form (15') are generated in Phase 1 instead of constraints of the form (15), the final solution of the U-BRP$_{\text{master}}$ obtained in Phase 1 is not proven to be optimal. In order not to lose optimality guarantee, all generated constraints of the form (15’) within Phase 1 are once again replaced by (15) and the procedure is continued by usage of (15). More specifically, if the result of Phase 1 is the U-BRP$_{\text{master}}$ extended by x generated constraints of the form (15’), then the input to Phase 2 is the U-BRP$_{\text{master}}$ extended by x corresponding constraints of the form (15). Thus, Phase 2 is an optimality check for the solution found in Phase 1. It is worth mentioning that in none of our instances, the optimal solution found at the end of Phase 2 was better than the feasible solution found at the end of Phase 1, and therefore, in all these cases Phase 1 already delivered an optimal solution of the original problem.

6 Computational study

In this section, the results of a computational study are presented. All model formulations and the row generation method are implemented in AMPL. The commercial solver Gurobi (version 9.1.1) is used to solve the models. The computational experiments are conducted on an Intel(R) Core(TM) i7-4770 CPU @ 3.4 GHz desktop computer with 16.0 GB RAM and Windows 10 Professional (64-bit).

The computational experiments are performed on the test (CV) instances proposed by Caserta et al. (2011) and Caserta et al. (2012). The authors randomly generated instances for different sizes of stacking areas, determined by the number of stacks $W \in \{3, 4, \ldots, 10\}$ and the maximal stack height $H \in \{5, 6, 7, 8, 12\}$. Each size of a stacking area defines an instance class (H-W) containing 40 instances. Any stack of the stacking area contains $H-2$ items, i.e., the two upper slots of each

Table 1 Comparison of the RG, the BRP–I and the BRP$_{m3}$

(H-W)	# Opt			Time (s)			Time* (s)		
	RG	BRP–I	m3	RG	BRP–I	m3	RG	BRP–I	m3
(5-3)	40	40	40	14.2	63.8	0.3	14.2	63.8	0.3
(5-4)	39	35	40	248.7	1027.3	4.6	162.7	961.3	2.1
(5-5)	39	22	40	262.1	2360.7	21.2	176.5	2328.9	11.4
(5-6)	37	3	39	526.9	3481.6	191.3	277.7	3472.0	84.8
(5-7)	33	0	36	945.8	3600.0	438.1	382.8	3600.0	187.1
(5-8)	27	0	33	1714.3	3600.0	969.0	806.4	3600.0	345.9
(6-4)	29	3	39	1395.2	3504.7	354.8	558.9	3468.6	53.8
(6-5)	18	0	24	2331.4	3600.0	1742.5	780.8	3600.0	360.0
(6-6)	9	0	16	3101.6	3600.0	2554.7	1384.9	3600.0	657.8
(6-7)	3	0	5	3531.4	3600.0	3402.8	2685.7	3600.0	2659.5
(7-4)	5	0	11	3290.4	3600.0	2705.7	1123.1	3600.0	42.8
(7-5)	0	0	2	3600.0	3600.0	3443.8	–	–	–

stack are empty. Thus, a (H-W) instance contains $N = W(H-2)$ items. The item numbers $1, \ldots, N$ are randomly allocated to the available slots of the stacking area. In total, the CV instances contain 21 instance classes and therefore a total of 840 test instances.

Within preliminary computational experiments, we compared the three proposed types of constraints (see Sect. 5.5). The results show no significant differences in performance for most test instances. Only for the largest test instances (5-6) of the preliminary study, Type 3 slightly outperforms Type 1 and 2 and therefore we decided to generate constraints of Type 3 within the computational study. Additionally, we make use of the tighter cuts (15') in Phase 1 of the procedure and (15) in Phase 2 (see Sect. 5.6). All proposed opportunities for fixing the variables (see Sect. 5.1 and 5.3), the valid inequalities (37'), (38'), (39), (40') and (41) and the symmetry breaking constraints (47) are applied.

Table 1 compares the performances of the new modeling approach (RG), its underlying basic model formulation BRP–I proposed by Caserta et al. (2012) and the state–of–the–art model formulation BRP_{m3} proposed by Lu et al. (2020). There is a need to calculate an upper and a lower bound on the total number of relocations, as there are parameters within the model formulations that require these values. Thus, we calculate an upper bound by means of a heuristic proposed by Caserta et al. (2012) and a lower bound called LB_3 proposed by Tricoire et al. (2018). The first column of Table 1 gives the instance classes. The columns two, three and four give the number of instances which could be optimally solved by the RG, the BRP–I and the BRP_{m3}, respectively. The columns five, six and seven give the average runtimes in seconds for the RG, the BRP–I and the BRP_{m3}, respectively, calculated over all instances of each class. Furthermore, the columns eight, nine and ten depict the respective average runtimes in seconds, calculated over the instances which could

Table 2 Instances for which the RG outperforms the BRP_{m3}

CV instance	(H-W)	# Reloc	Mean # reloc	Time (s)	
				RG	m3
88	(5-5)	9	6.7	28.4	36.4
127	(5-6)	11	8.1	135.5	1088.5
142	(5-6)	10	8.1	230.2	794.3
179	(5-7)	11	8.7	216.3	276.9
198	(5-7)	11	8.7	345.4	3600.0
205	(5-8)	12	9.6	2461.9	3600.0
233	(5-8)	11	9.6	657.1	1635.8
259	(6-4)	11	9.0	106.7	111.5
276	(6-4)	11	9.0	305.8	628.6
286	(6-5)	12	10.4	330.1	656.5
318	(6-5)	12	10.4	565.1	3450.3
359	(6-6)	13	10.3	1410.2	3600.0
371	(6-7)	15	13.0	2981.6	3600.0
398	(6-7)	13	13.0	2441.1	3600.0

be optimally solved by the RG. A time limit of 3600 s for each instance is implemented. Instances larger than (H-W) = (7-5) are not presented since none could be optimally solved within the time limit.

The results given in Table 1 confirm that the model BRP_{m3} outperforms the model BRP–I. Furthermore, the results show that the RG outperforms the model formulation BRP–I. However, the difference in performance between the two model formulations BRP–I and BRP_{m3} is quite large, so that although the RG framework greatly increases the efficiency of the underlying model BRP–I, it cannot yet compete with the model BRP_{m3}.

Nevertheless, it is worth noting that 14 of the instances are solved faster by the RG as by the BRP_{m3} and all these instances have an optimal solution value (number of relocations) that is greater than or equal to the average of the corresponding instance class. Details on this are presented in Table 2. The first and second columns give the CV instance number and the corresponding instance class, respectively. The third column presents the optimal numbers of relocations for the specific instances. The column four indicates the average optimal numbers of relocations for the respective instance classes. Finally, the columns five and six give the runtimes in seconds used by the RG and the BRP_{m3}, respectively, for solving the corresponding CV instance. For example, the CV instance # 359 (instance class 6-6) requires 13 relocations in an optimal solution and can be optimally solved by the RG in 1410.2 s, while the BRP_{m3} hits the time limit of 3600 s. Thus, although the BRP_{m3} performs better than the RG on average, the RG may have advantages on specific instances, especially when an optimal solution requires a relatively large number of relocations. This may indicate that a restriction to allow at most one relocation per transition within a mathematical model (like in the BRP–I and the BRP_{m3}) is a

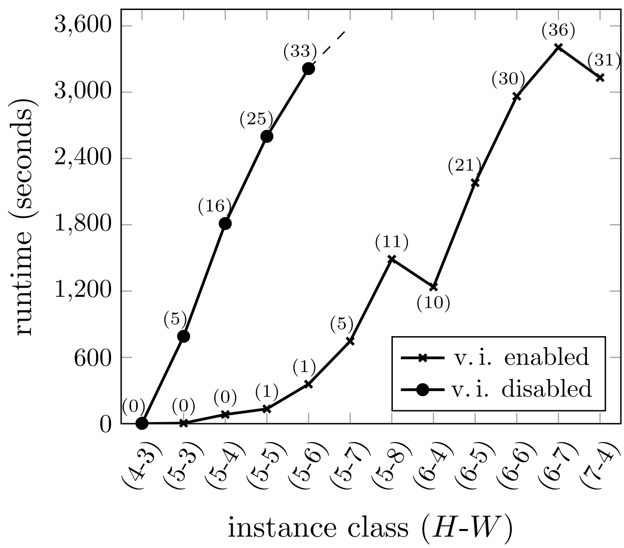

Fig. 10 Average runtimes with valid inequalities enabled/disabled. The numbers within the brackets denote the numbers of instances running into the time limit of 3600 s, respectively

Table 3 Results for the new modeling approach (RG)

(H-W)	# Solved		Time* (s)			Median time* (s)
	P1	P1 &2	P1	P2	P1 &2	
(5-3)	40	40	4.5	9.7	14.2	1.6
(5-4)	40	39	68.4	94.3	162.7	6.9
(5-5)	39	39	44.6	131.9	176.5	28.2
(5-6)	39	37	179.3	98.4	277.7	100.6
(5-7)	35	33	221.8	161.0	382.8	155.2
(5-8)	29	27	546.8	259.6	806.4	598.8
(6-4)	30	29	343.4	215.5	558.9	127.1
(6-5)	19	18	532.8	248.0	780.8	566.8
(6-6)	10	9	930.4	454.5	1384.9	1376.5
(6-7)	4	3	1977.8	707.9	2685.7	2634.4
(7-4)	9	5	1023.4	99.7	1123.1	921.5
(7-5)	0	0	–	–	–	–

limiting factor. Thus, a relaxation of this restriction has high potential to speed up the solution time.

The valid inequalities (see Sect. 5.2) improve the solution quality of the U-BRP$_{master}$ and therefore reduces the number of iterations required, but since most of the valid inequalities exploit a Big–M, the time for solving the U-BRP$_{master}$ tends to increase. Therefore, the impact of the valid inequalities are investigated by further computational experiments. Figure 10 depicts the results of the comparison of the procedure with the valid inequalities enabled and disabled. The numbers within the brackets indicate the respective numbers of instances that ran into the time limit of 3600 s. The settings are like in the main study, with the exception that only Phase 1 is investigated. Additionally, we generated 40 further instances with (H-W) = (4-3) and N = 6 randomly allocated items. It can be observed that the valid inequalities significantly reduce the runtimes. Without applying the valid inequalities, the number of instances which cannot be solved increases very strongly by extending the instance size. For (H-W) = (5-6), e.g., the major part (33 of 40) could no longer be solved. None of the instances greater than (5-6) are solved within the time limit unless the valid inequalities are applied.

Finally, Table 3 provides more detailed results of the new modeling approach. The first column of Table 3 again gives the instance classes. The second and third columns present the number of instances for which the first phase and both phases could be finished, respectively. Remember that a solution provided by Phase 1 is feasible for the U-BRP but only proven to be optimal after finishing Phase 2. In this context, it is worth mentioning that for the instances which are optimally solved (proven by finishing Phase 2), all solutions obtained after Phase 1 were already optimal. Thus, in these cases, the initial solution in Phase 2 is already optimal with respect to the original problem, even though it is not yet proven at this point. Already having an optimal solution for the original problem is obviously a great initial situation for the optimization procedure within Phase 2 and therefore Phase 2 has shorter average runtimes compared to Phase 1 in many cases. This can be observed in the columns four, five and six which give the average runtimes for the completion of Phase 1, Phase 2 and the whole procedure, respectively. The last column states the *median* of the runtimes for the RG which are lower than the average runtimes in all cases, in most cases even significantly lower. The measures presented in Table 3 are computed over the instances which could be optimally solved by the RG.

Even though the new modeling approach cannot yet compete with the model formulation BRP_{m3} from Lu et al. (2020) or the most effective search-based methods, e.g., the branch–and–bound algorithms from Jin and Tanaka (2023), Tanaka and Mizuno (2018) and Tricoire et al. (2018), it gives rise to spend more effort on model-based approaches with a focus on the reduction of occurring configurations.

7 Conclusion

This paper adresses the unresticted block relocation problem (U-BRP). A mathematical model formulation is developed to enable multiple moves within each transition, which is prohibited in most models from the literature. This significantly decreases the size of such model formulations and may speed up the solution time. The proposed model formulation is embedded within a row generation framework to construct an optimal solution for the U-BRP. The results of the computational study show that this modeling approach outperforms the underlying basic model formulation.

Although there are more effective procedures for solving the U-BRP optimally, e.g., the branch–and–bound algorithms developed by Jin and Tanaka (2023), Tanaka and Mizuno (2018) and Tricoire et al. (2018), or the model formulation proposed by Lu et al. (2020), the results of the computational study show that the proposed modifications can significantly speed up model-based approaches. Like in the literature concerning the R-BRP, search-based methods have been considered to be superior to model-based approaches. Nevertheless, a recent publication provides a very competitive model-based branch–and–cut approach for the R-BRP.

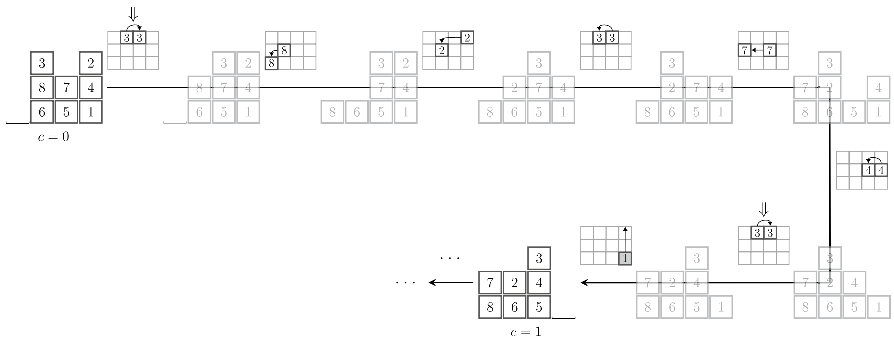

Fig. 11 Feasible solution of the $\text{U-BRP}_{\text{master}}$ with two identical relocations of the same item

Future research on model-based approaches for the U-BRP could focus on reducing the number of variables by means of a reduction of the number of configurations that are considered, i.e., enable multiple moves within each transition. With respect to the proposed row generation framework, it could be worth to spend more effort on developing techniques for a further reduction of the number of iterations required. A further and probably more promising approach might be to make use of an adapted modification of a better performing model formulation, e.g., the BRP_{m2} or the BRP_{m3}, and embed it within the row generation framework.

Appendix 1: Feasible solution with two identical relocations of the same item

Figure 11 depicts a feasible solution of the $\text{U-BRP}_{\text{master}}$ (and the original problem) which contains two identical relocations. These are the relocations of item 3 from *slot* (2, 3) to *slot* (3, 3) which take place twice and therefore the corresponding decision variable $\bar{x}_{233331}$ has the value 2. Note that this instance can be solved with 6 relocations and therefore the depicted solution is not optimal but only feasible. We have not found any instance with an optimal solution containing an identical relocation of the same item more than once. We even expect such solutions not to exist, but since this is not proven, we follow a cautious approach in order not to falsely cut off such an optimal solution and therefore the variables $\bar{x}_{ijklnc}$ are defined as integer variables ($\bar{x}_{ijklnc} \in \mathbb{N}_0$). For a better readability, the configurations which are actually omitted within the $\text{U-BRP}_{\text{master}}$ are presented in light gray.

Appendix 2: The weights within the penalty term must be at least $\gamma_{\bar{c}}$

Remember the objective (17) ($\max \sum_{c=1}^{\gamma_{\bar{c}}} (v_{\bar{c}c} - \gamma_{\bar{c}} \lambda_c)$) of the $\text{U-BRP}_{\text{sub}(\bar{c})}$ and the fact that the number of s-transitions always equals the number of moves $\gamma_{\bar{c}}$ contained in the $\text{U-BRP}_{\text{sub}(\bar{c})}$ (or in the respective m-transition). We can distinguish two cases which may arise:

(i) A feasible solution of the $\text{U-BRP}_{\text{sub}(\bar{c})}$ exists such that $\lambda_c = 0$ for all $c = 1, \dots, \gamma_{\bar{c}}$. The smallest possible optimal objective function value that can exist for such solutions is 1. This is a solution with one move in each s-transition (determined by $\lambda_c = 0, c = 1, \dots, \gamma_{\bar{c}}$) and therefore all penalty terms have the value 0 ($\Rightarrow \sum_{c=1}^{\gamma_{\bar{c}}} \gamma_{\bar{c}} \lambda_c = 0$) whereby the retrieval takes place in the last s-transition ($\Rightarrow \sum_{c=1}^{\gamma_{\bar{c}}} v_{\bar{c}c} = 1$). If the retrieval takes place in an earlier s-transition, the objective function value increases.

(ii) No feasible solution of the $\text{U-BRP}_{\text{sub}(\bar{c})}$ exists such that $\lambda_c = 0$ for all $c = 1, \dots, \gamma_{\bar{c}}$ and therefore there is at least one $\lambda_c > 0$. In this case, the greatest possible optimal objective function value is 0. The smallest value that the sum of all penalty terms can have in this case arises in a solution with exactly one $\lambda_c = 1$ and all others having the value 0 ($\Rightarrow \sum_{c=1}^{\gamma_{\bar{c}}} \gamma_{\bar{c}} \lambda_c = \gamma_{\bar{c}}$). Thus, it is a solu-

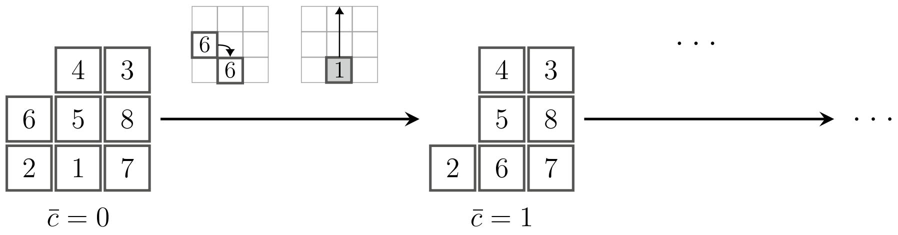

Fig. 12 First m-transition of an optimal solution of the U-BRP$_{\text{master}}$ (iteration 1)

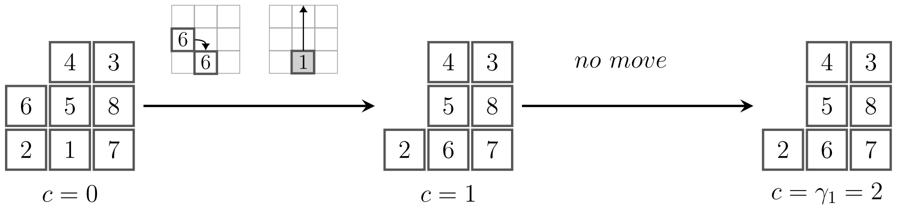

Fig. 13 Optimal solution of the U-BRP$_{\text{sub}(1)}$ (iteration 1)

tion with one s-transition containing two moves, one s-transition containing no move and all other s-transitions containing exactly one move. Furthermore, the term $\sum_{c=1}^{\gamma_{\bar{c}}} v_{\bar{c}c}$ has its maximum value if the retrieval takes place in the first s-transition ($\Rightarrow \sum_{c=1}^{\gamma_{\bar{c}}} v_{\bar{c}c} = \gamma_{\bar{c}}$). In total, such solutions cannot have a greater optimal objective function value than $\gamma_{\bar{c}} - \gamma_{\bar{c}} = 0$.

Thus, a feasible solution of the U-BRP$_{\text{sub}(\bar{c})}$ with at least one $\lambda_c > 0$ is never optimal if there exists a feasible solution with $\lambda_c = 0$ for all $c = 1, \ldots, \gamma_{\bar{c}}$, i.e., if an optimal solution of the U-BRP$_{\text{sub}(\bar{c})}$ has at least one $\lambda_c > 0$, then there is no feasible solution with $\lambda_c = 0$ for all $c = 1, \ldots, \gamma_{\bar{c}}$.

Appendix 3: Discussion on the constraints (23)

3.1: Example which requires the constraints (23)

Given an instance with $W = 3, H = 5, N = 8$ and the initial m-configuration given in Fig. 12 at $\bar{c} = 0$. Figure 12 presents an initial optimal solution of the U-BRP$_{\text{master}}$, i.e., no constraints are generated yet. Note that only the first m-transition is of interest and therefore presented.

The corresponding subproblem U-BRP$_{\text{sub}(1)}$ has the optimal solution presented in Fig. 13.

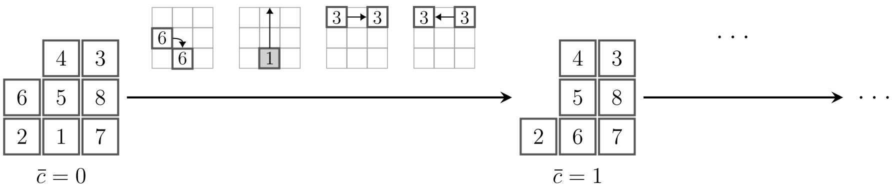

Fig. 14 First m-transition of an optimal solution of the U-BRP$_{\text{master}}$ (iteration 57)

Since there is no sequence for the two moves which leads to a feasible s-configuration after each single move, this combination of moves is prohibited by adding the constraint

$$\begin{aligned}\bar{x}_{122161} + \bar{y}_{2111} \le 1 + \sum_{(i,j,k,l,n)\in X\setminus\{(1,2,2,1,6)\}} \bar{x}_{ijkln1} \\ + \sum_{\substack{(i,j,n)\,\in\, Y\setminus\{(1,1,2),(1,2,6),(2,1,1),\\ (2,2,5),(2,3,4),(3,1,7),\\ (3,2,8),(3,3,3)\}}} \bar{b}_{ijn0} \\ + \sum_{\substack{(i,j,n)\,\in\, Y\setminus\{(1,1,2),(2,1,6),(2,2,5),\\ (2,3,4),(3,1,7),(3,2,8),\\ (3,3,3)\}}} \bar{b}_{ijn1}\end{aligned}$$

to the U-BRP$_{\text{master}}$. From now on, these two moves can only take place together within the first m-transition of the U-BRP$_{\text{master}}$ if at least one further relocation is added to this m-transition (or if the adjacent m-configurations differ). In iteration 57 this scenario occurs and the two moves from iteration 1 take place in an extended combination of moves (Fig. 14).

It is obvious that the two relocations of item 3 are superfluous since no item located below item 3 is relocated or retrieved. The only reason for the "back and forth"-relocations of item 3 is that it enables the moves from iteration 1 ($\bar{x}_{122161}$ and $\bar{y}_{2111}$) to take place once again within the first m-transition.

These four moves of the first m-transition should now take place in the resulting subproblem U-BRP$_{\text{sub}(1)}$, but without constraints (23), the optimal solution of the resulting subproblem U-BRP$_{\text{sub}(1)}$ contains only a subset of the moves and not all.

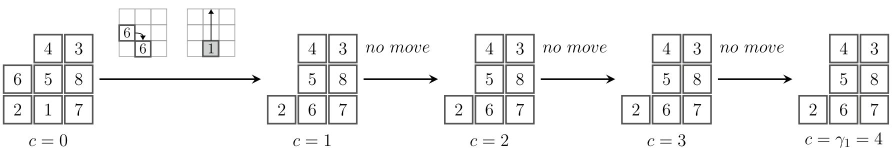

Fig. 15 Optimal solution of the U-BRP$_{\text{sub}(1)}$ (iteration 57)

The optimal solution (Fig. 15) is very similar to that from iteration 1. The only difference is that $\gamma_1 = 4$ s-configurations are considered.

Thus, the constraint of iteration 1 is generated again which does not affect the solution space. From now on, the optimal solution of the U-BRP$_{\text{master}}$ remains the same and thus the procedure does not terminate.

3.2: The procedure never causes a deadlock if the constraints (23) are Enabled

In general, a row generation procedure never causes a deadlock if (i) the set of feasible solutions of the relaxed problem is finite and (ii) at least one feasible solution is cut off from the solution space of the relaxed problem per iteration.

Let $\boldsymbol{x}^*$ be a feasible solution of the U-BRP$_{\text{master}}$ which is optimal with respect to the original problem U-BRP. Furthermore, let $\bar{\boldsymbol{X}}$ be the set of all feasible solutions of the U-BRP$_{\text{master}}$. For any solution $\boldsymbol{x} \in \bar{\boldsymbol{X}}$, let $r(\boldsymbol{x})$ be the corresponding number of relocations. Furthermore, let $\bar{\boldsymbol{X}}^r := \{\boldsymbol{x} \in \bar{\boldsymbol{X}} : r(\boldsymbol{x}) \leq r(\boldsymbol{x}^*)\}$ be the set of all feasible solutions of the U-BRP$_{\text{master}}$ with at most $r(\boldsymbol{x}^*)$ relocations.

Condition (i) is fulfilled: Due to the fact that the stacking area (and therefore the number of possible relocations) is limited by the number W of stacks, the height limit H and the number N of items, the set $\bar{\boldsymbol{X}}^r$ is finite. In case of an unlimited maximal height, without loss of generality, $H = N$ can be assumed since this is the maximal stacking height which can be obtained.

Condition (ii) is fulfilled: The constraints (23) ensure that each relocation of the $\bar{c}$-th m-transition takes place within the solution of the U-BRP$_{\text{sub}(\bar{c})}$. Thus, if the U-BRP$_{\text{sub}(\bar{c})}$ detects infeasibility, then all moves contained in the U-BRP$_{\text{sub}(\bar{c})}$ (i.e., contained in the $\bar{c}$-th m-transition) are involved in a newly generated constraint of the form (15). This cuts off at least one feasible solution $\boldsymbol{x} \in \bar{\boldsymbol{X}}^r$ from $\bar{\boldsymbol{X}}^r$.

Note that condition (ii) is also fulfilled for cuts of Type 2 and Type 3 presented in Sect. 5.5. These types do not involve all moves contained in the U-BRP$_{\text{sub}(\bar{c})}$ in a newly generated constraint, but all moves which cause infeasibility. Thus, there is also at least one feasible solution $\boldsymbol{x} \in \bar{\boldsymbol{X}}^r$ being cut off from $\bar{\boldsymbol{X}}^r$.

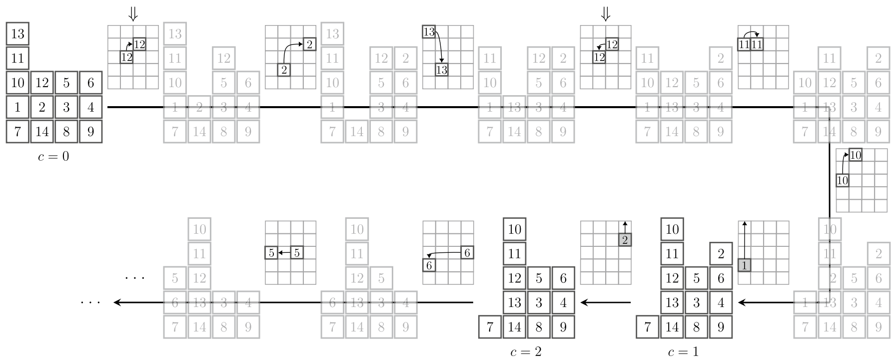

Fig. 16 Feasible solution of the U-BRP$_{\text{master}}$ with cyclic relocations which is optimal with respect to the original problem

Table 4 Exemplary partial solutions of the U-BRP$_{\text{master}}$ and their feasibility with respect to the different types of generated constraints

Exemplary partial solutions		Type 1	Type 2	Type 3
1	$\bar{x}_{132151} = 1, \bar{x}_{211331} = 1, \bar{x}_{231161} = 1, \bar{x}_{132331} = 1, \bar{y}_{1111} = 1$	Infeas	Infeas	Infeas
2	$\bar{x}_{132151} = 1, \bar{x}_{211331} = 1, \bar{x}_{231161} = 1, \bar{y}_{1111} = 1$	Feas	Infeas	Infeas
3	$\bar{x}_{132151} = 1, \bar{x}_{211331} = 1$	Feas	Feas	Infeas
4	$\bar{x}_{231161} = 1, \bar{y}_{1111} = 1$	Feas	Feas	Infeas

Since $\boldsymbol{x}^* \in \bar{\boldsymbol{X}}^r$ and $|\bar{\boldsymbol{X}}^r| < \infty$, the worst case scenario is a complete enumeration of all feasible solutions of the U-BRP$_{\text{master}}$ with at most $r(\boldsymbol{x}^*)$ relocations, i.e., a complete calculation of the set $\bar{\boldsymbol{X}}^r$ for finding $\boldsymbol{x}^*$.

Appendix 4: Instance with mandatory cyclic relocations

Figure 16 depicts an instance which has no optimal solution free of cyclic relocations, i.e., an item must be relocated at least twice within an m-transition c whereby it is relocated to the slot in which it was initially located in the originating m-configuration $c - 1$. The solution presented in Fig. 16 is optimal with respect to the original problem U-BRP and is depicted within the framework of the U-BRP$_{\text{master}}$. The relocations of item 12 within the first m-transition are of interest in this example. The optimal solution contains 8 relocations. If any relocation of item 12 ending in its initial *slot* (2, 3) is prohibited within the first m-transition, then there does not exist anymore a solution with 8 relocations and therefore cyclic relocations of item 12 are mandatory. For a better readability, the configurations which are actually omitted within the U-BRP$_{\text{master}}$ are presented in light gray.

Appendix 5: Further clarification that constraints of type 3 are the tightest constraints

The following example is a recap of the same example presented in Sect. 5.5. Initially, the first m-transition contains 4 relocations and 1 retrieval defined by the partial solution $\bar{x}_{132151} = 1, \bar{x}_{211331} = 1, \bar{x}_{231161} = 1, \bar{x}_{132331} = 1, \bar{y}_{1111} = 1$. All other $\bar{x}$– and $\bar{y}$ –variables within the first m-transition have the value 0 in this example as well as in all following exemplary partial solutions (all $\bar{x}_{ijkln1}$ and $\bar{y}_{ijn1}$ have the value 0). Thus, for a better readibility, all decision variables on the right–hand side can be omitted. The three different types of constraints generated by the procedure are as follows:

$$\text{Type 1:} \quad \bar{x}_{132151} + \bar{x}_{211331} + \bar{x}_{231161} + \bar{x}_{132331} + \bar{y}_{1111} \leq 4$$
$$\text{Type 2:} \quad \bar{x}_{132151} + \bar{x}_{211331} + \bar{x}_{231161} + \bar{y}_{1111} \leq 3$$
$$\text{Type 3:} \quad \bar{x}_{132151} + \bar{x}_{211331} \leq 1 \quad \text{and} \quad \bar{x}_{231161} + \bar{y}_{1111} \leq 1$$

Given the constraints of the different types, the following Table 4 gives some exemplary partial solutions. Furthermore, it indicates whether the solutions are feasible or infeasible (and thus cut off from the solution space) with respect to the constraints of Type 1, 2 and 3, respectively.

⇒ Type 2 constraints cut off all (partial) solutions which are cut off by Type 1 constraints (solution 1 in this example). Additionally, Type 2 constraints cut off further (partial) solutions which are not cut off by Type 1 constraints (solution 2 in this example). Thus, Type 2 constraints are tighter than Type 1 constraints.

⇒ Type 3 constraints cut off all (partial) solutions which are cut off by Type 2 constraints (solutions 1 and 2 in this example). Additionally, Type 3 constraints cut off further (partial) solutions which are not cut off by Type 2 constraints (solutions 3 and 4 in this example). Thus, Type 3 constraints are tighter than Type 2 constraints.

Funding Open Access funding enabled and organized by Projekt DEAL. We acknowledge support by the Open Access Publication Fund of the University of Duisburg-Essen.

Data availability The datasets generated and analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

Bacci T, Mattia S, Ventura P (2019) The bounded beam search algorithm for the block relocation problem. Comput Oper Res 103:252–264. https://doi.org/10.1016/j.cor.2018.11.008

Bacci T, Mattia S, Ventura P (2020) A branch-and-cut algorithm for the restricted block relocation problem. Eur J Oper Res 287(2):452–459. https://doi.org/10.1016/j.ejor.2020.05.029

Caserta M, Voß S, Sniedovich M (2011) Applying the corridor method to a blocks relocation problem. OR Spectr 33(4):915–929. https://doi.org/10.1007/s00291-009-0176-5

Caserta M, Schwarze S, Voß S (2012) A mathematical formulation and complexity considerations for the blocks relocation problem. Eur J Oper Res 219(1):96–104. https://doi.org/10.1016/j.ejor.2011.12.039

Caserta M, Schwarze S, Voß S (2009) A new binary description of the blocks relocation problem and benefits in a look ahead heuristic. In: Cotta C, Cowling PI (eds) Evolutionary computation in combinatorial optimization. Lecture notes in computer science, vol. 5482, pp. 37–48. Springer, Berlin. https://doi.org/10.1007/978-3-642-01009-5_4

Caserta M, Schwarze S, Voß S (2011) Container rehandling at maritime container terminals. In: Böse JW (ed) Handbook of terminal planning. Operations research/computer science interfaces series, vol. 49, pp. 247–269. Springer, New York, NY. https://doi.org/10.1007/978-1-4419-8408-1_13

Covic F (2019) Container handling in automated yard blocks: an integrative approach based on time information. Contributions to management science. Springer, Cham. https://doi.org/10.1007/978-3-030-05291-1

de Melo da Silva M, Toulouse S, Wolfler Calvo R (2018) A new effective unified model for solving the pre-marshalling and block relocation problems. Eur J Oper Res 271(1):40–56. https://doi.org/10.1016/j.ejor.2018.05.004

Expósito-Izquierdo C, Melián-Batista B, Marcos Moreno-Vega J (2014) A domain-specific knowledge-based heuristic for the blocks relocation problem. Adv Eng Inform 28(4):327–343. https://doi.org/10.1016/j.aei.2014.03.003

Expósito-Izquierdo C, Melián-Batista B, Moreno-Vega JM (2015) An exact approach for the blocks relocation problem. Expert Syst Appl 42(17–18):6408–6422. https://doi.org/10.1016/j.eswa.2015.04.021

Forster F, Bortfeldt A (2012) A tree search procedure for the container relocation problem. Comput Oper Res 39(2):299–309. https://doi.org/10.1016/j.cor.2011.04.004

Galle V, Barnhart C, Jaillet P (2018) A new binary formulation of the restricted container relocation problem based on a binary encoding of configurations. Eur J Oper Res 267(2):467–477. https://doi.org/10.1016/j.ejor.2017.11.053

Jin B, Tanaka S (2023) An exact algorithm for the unrestricted container relocation problem with new lower bounds and dominance rules. Eur J Oper Res 304(2):494–514. https://doi.org/10.1016/j.ejor.2022.04.006

Jin B, Zhu W, Lim A (2015) Solving the container relocation problem by an improved greedy look-ahead heuristic. Eur J Oper Res 240(3):837–847. https://doi.org/10.1016/j.ejor.2014.07.038

Jovanovic R, Voß S (2014) A chain heuristic for the blocks relocation problem. Comput Ind Eng 75:79–86. https://doi.org/10.1016/j.cie.2014.06.010

Jovanovic R, Tanaka S, Nishi T, Voß S (2019) A grasp approach for solving the blocks relocation problem with stowage plan. Flex Serv Manuf J 31(3):702–729. https://doi.org/10.1007/s10696-018-9320-3

Kim KH, Hong G-P (2006) A heuristic rule for relocating blocks. Comput Oper Res 33(4):940–954. https://doi.org/10.1016/j.cor.2004.08.005

Ku D, Arthanari TS (2016) On the abstraction method for the container relocation problem. Comput Oper Res 68:110–122. https://doi.org/10.1016/j.cor.2015.11.006

Lee Y, Hsu N-Y (2007) An optimization model for the container pre-marshalling problem. Comput Oper Res 34(11):3295–3313. https://doi.org/10.1016/j.cor.2005.12.006

Lehnfeld J, Knust S (2014) Loading, unloading and premarshalling of stacks in storage areas: survey and classification. Eur J Oper Res 239(2):297–312. https://doi.org/10.1016/j.ejor.2014.03.011

Lu C, Zeng B, Liu S (2020) A study on the block relocation problem: lower bound derivations and strong formulations. IEEE Trans Autom Sci Eng 17(4):1829–1853. https://doi.org/10.1109/TASE.2020.2979868

Petering MEH, Hussein MI (2013) A new mixed integer program and extended look-ahead heuristic algorithm for the block relocation problem. Eur J Oper Res 231(1):120–130. https://doi.org/10.1016/j.ejor.2013.05.037

Quispe KEY, Lintzmayer CN, Xavier EC (2018) An exact algorithm for the blocks relocation problem with new lower bounds. Comput Oper Res 99:206–217. https://doi.org/10.1016/j.cor.2018.06.021

Stahlbock R, Voß S (2008) Operations research at container terminals: a literature update. OR Spectr 30(1):1–52. https://doi.org/10.1007/s00291-007-0100-9

Steenken D, Voß S, Stahlbock R (2004) Container terminal operation and operations research—a classification and literature review. OR Spectr 26(1):3–49. https://doi.org/10.1007/s00291-003-0157-z

Tanaka S, Mizuno F (2018) An exact algorithm for the unrestricted block relocation problem. Comput Oper Res 95:12–31. https://doi.org/10.1016/j.cor.2018.02.019

Tanaka S, Takii K (2016) A faster branch-and-bound algorithm for the block relocation problem. IEEE Trans Autom Sci Eng 13(1):181–190. https://doi.org/10.1109/TASE.2015.2434417

Tanaka S, Voß S (2022) An exact approach to the restricted block relocation problem based on a new integer programming formulation. Eur Jo Oper Res 296(2):485–503. https://doi.org/10.1016/j.ejor.2021.03.062

Tanaka S, Mizuno F (2015) Dominance properties for the unrestricted block relocation problem and their application to a branch-and-bound algorithm. In: 2015 IEEE international conference on automation science and engineering (CASE), pp. 509–514. IEEE, Gothenburg. https://doi.org/10.1109/CoASE.2015.7294130

Tang L, Jiang W, Liu J, Dong Y (2015) Research into container reshuffling and stacking problems in container terminal yards. IIE Trans 47(7):751–766. https://doi.org/10.1080/0740817X.2014.971201

Ting C-J, Wu K-C (2017) Optimizing container relocation operations at container yards with beam search. Trans Res Part E Logist Transp Rev 103:17–31. https://doi.org/10.1016/j.tre.2017.04.010

Tricoire F, Scagnetti J, Beham A (2018) New insights on the block relocation problem. Comput Oper Res 89:127–139. https://doi.org/10.1016/j.cor.2017.08.010

Ünlüyurt T, Aydın C (2012) Improved rehandling strategies for the container retrieval process. J Adv Transp 46(4):378–393. https://doi.org/10.1002/atr.1193

Wan Y-W, Liu J, Tsai P-C (2009) The assignment of storage locations to containers for a container stack. Naval Res Logist 56(8):699–713. https://doi.org/10.1002/nav.20373

Zehendner E, Feillet D (2014) A branch and price approach for the container relocation problem. Int J Prod Res 52(24):7159–7176. https://doi.org/10.1080/00207543.2014.965358

Zehendner E, Caserta M, Feillet D, Schwarze S, Voß S (2015) An improved mathematical formulation for the blocks relocation problem. Eur J Oper Res 245(2):415–422. https://doi.org/10.1016/j.ejor.2015.03.032

Zhang C, Guan H, Yuan Y, Chen W, Wu T (2020) Machine learning-driven algorithms for the container relocation problem. Trans Res Part B Methodol 139:102–131. https://doi.org/10.1016/j.trb.2020.05.017

Zhang H, Guo S, Zhu W, Lim A, Cheang B (2010) An investigation of ida* algorithms for the container relocation problem. In: García-Pedrajas N, Herrera F, Fyfe C, Benítez Sánchez JM, Ali M (eds) Trends in applied intelligent systems. Lecture notes in artificial intelligence, vol. 6096, pp. 31–40. Springer, Berlin. https://doi.org/10.1007/978-3-642-13022-9_4

Zhu W, Qin H, Lim A, Zhang H (2012) Iterative deepening a* algorithms for the container relocation problem. IEEE Trans Autom Sci Eng 9(4):710–722. https://doi.org/10.1109/TASE.2012.2198642