The Load Planning and Sequencing Problem for Double-Stack Intermodal Trains

Moritz Ruf[a,*], Jean-François Cordeau[b,c], Emma Frejinger[d,c]

[a] *Technische Universität Dresden, Institute of Railway Systems and Public Transport, 01062 Dresden, Germany*
[b] *HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, H3T 2A7 Canada*
[c] *CIRRELT, Pavillon André Aisenstadt, 2920 chemin de la Tour, Montréal, H3T 1J4 Canada*
[d] *Department of Computer Science and Operations Research, Université de Montréal, Pavillon André Aisenstadt, 2920 chemin de la Tour, Montréal, H3T 1J4 Canada*

Abstract

This paper addresses the integrated load planning and sequencing problem for double-stack intermodal trains. This decision problem occurs in intermodal terminals and consists in assigning containers from a storage area to slots on railcars of outbound trains and in determining the loading sequence of the handling equipment. Even though this operational problem is relevant in intermodal terminals, it has seen no attention in the operations research literature so far. Prior models either focus on single-stack railcars or treat the load planning and the load sequencing problems separately. By extending prior work on load planning, we propose four integer linear programming formulations differing in the number of constraints and variables. An extensive numerical study identifies two formulations that perform best in our setting with respect to the number of optimal solutions found in a given time limit and average solution time. With these formulations, we solve medium-size instances related to one block of an intermodal train with a commercial general-purpose solver in less than 30 minutes. A case study based on real data from the North American market provided by the Canadian National Railway Company highlights that the integrated load planning and sequencing problem can reduce the number of container handlings in intermodal terminals compared to sequential solutions by on average 11.3% and 16.5% for gantry cranes and reach stackers, respectively.

Keywords: Transportation; freight; intermodal railway terminals; double-stack train loading; load sequencing

1. Introduction

Intermodal transportation plays an important role in global supply chains and is a growing market. In this context, load units are transported from origin to destination using at least two different modes of transportation. When the long-haul leg of ground transportation is carried out by rail, the load units – usually in the form of standardized containers – must

*Corresponding author
Email addresses: moritz.ruf@tu-dresden.de (Moritz Ruf), jean-francois.cordeau@hec.ca (Jean-François Cordeau), emma.frejinger@cirrelt.ca (Emma Frejinger)

be transferred from one transportation mode to another. This takes place in intermodal rail terminals, which are important to the efficiency of the overall transport chain.

In these terminals, containers are temporarily piled in stacks awaiting to be loaded onto outbound railcars. Even though containers are standardized load units, they come in different types and have individual characteristics, such as weight. Similarly, there exists a variety of railcar types, each characterized, e.g., by the number of platforms, the length of the platforms and whether they can be double or single stacked. Handling equipment transfers containers from the stacks and loads them onto railcars. There are different kinds of equipment: vehicles (e.g., reach stackers [RS]) and cranes (e.g., gantry cranes [GC]). The characteristics of the equipment, such as reach and lifting capacity, influence the set of containers that are directly accessible at any given point in time.

This paper addresses an operational planning problem which consists in selecting a set of containers to be loaded, assigning these containers to specific positions on double-stack railcars (so-called slots) and determining the order in which the containers are retrieved from the stack and placed on the railcars so as to maximize the use of the railcars and minimize the handling effort in the terminal. We refer to this problem as the load planning and sequencing problem (LPSP). Since we do not decide on the time, but only on the order of the movements, we employ sequencing as opposed to scheduling. We consider a static and deterministic case where all the information about a given problem instance is perfectly known before the loading starts and it does not change during the loading process. We solve the LPSP for one handling equipment and one block of an intermodal train, where a block is a consolidation of railcars and containers with the same destination (e.g., Ahuja et al., 2007; Morganti et al., 2019).

Two observations of the double-stack load planning problem (LPP) – assignment of containers to positions on railcars – (Mantovani et al., 2018) highlight the importance of the LPSP and associated challenges. First, there can be many optimal solutions to the LPP, each leading to different load sequence solutions. Consequently, solving the two problems jointly, as opposed to first solving the LPP and keeping the corresponding solution fixed when optimizing the loading sequence, may lead to reduced handling costs. Second, the LPP crucially depends on, e.g., the weight of individual containers as there are constraints on the centre of mass. This detailed representation makes the LPSP challenging to model and to solve in reasonable time (in our case 30 minutes).

The LPSP is related to other tactical and operational planning problems occurring in intermodal terminals: Closest are the planning of the terminal layout (Boysen et al., 2013), the storage space allocation problem (Zhang et al., 2003) and the scheduling of cranes (Bierwirth and Meisel, 2010). In this context, we assume a fixed terminal layout where containers are stored according to their block, i.e., their expected departure date and destination. A train is composed of a sequence of blocks. Containers that are not loaded on their expected block need to wait for the following train departure. Consistent with the reality in many terminals, the LPSP can hence be decomposed by block and handling equipment, making our assumptions acceptable.

Most of the existing literature related to the LPP considers single-stack railcars (e.g., Bruns and Knust, 2012; Dotoli et al., 2013). However, for the North American market double stacking of containers is of high relevance. Mantovani et al. (2018) propose a model for the LPP dealing with a high variety of containers and railcars, including double-stacking. They solve instances with up to over 1,000 containers to optimality using a generic all-purpose solver. However, the load sequencing is not considered as part of the problem. A few papers have addressed the single-stack LPSP, most noteworthy are Corry and Kozan (2006) and

Ambrosino et al. (2011). Even though they consider the simpler single stack case, large instances cannot be solved by exact methods. Part of the models are based on simplifying assumptions such as pre-set loading sequences, the exclusion of rehandling movements, or a fixed set of containers to load (i.e., no overbooking). The largest instances solved to optimality without those simplifying assumptions comprise 40 containers (Ambrosino et al., 2013).

Contributions. The paper makes three main contributions:

1. We introduce the LPSP for double-stack intermodal trains along with four integer linear program (ILP) formulations adaptable to model the restrictions related to handling equipment and terminal layout. This problem is of high practical relevance, especially for the North American market, but has seen no attention in the literature so far. We hereby build on the work of Mantovani et al. (2018) and extend their mathematical model. The presented models differ in the number of constraints and variables and they may perform differently depending on instance characteristics. This gives interesting insights from a theoretical point of view and allows practitioners to choose the formulation that works best in their particular environment.

2. Based on realistic data from one of the largest railroads in North America, we assess the performance of the ILP formulations by conducting extensive numerical experiments using a general purpose ILP solver (IBM CPLEX). While the terminal layout is designed for RS vehicles, we illustrate the generality of the formulations by also reporting results for a GC. In addition, we evaluate different ways to integrate distance in the objective function enabling to solve larger instances while demonstrably maintaining a high solution quality. The best performing formulations allow to find optimal solutions for instances of 75 containers.

3. We report results comparing the LPSP solutions to those obtained by a sequential solution approach, i.e., optimizing the load sequence for a fixed LPP solution. In the case of instances comprising 50 containers and the distance cost for loaded moves, we note an important reduction in the average number of container moves: 11.3% and 16.5% for the GC and RS, respectively.

The remainder of this paper is structured as follows: Section 2 gives an overview of the related literature. Section 3 introduces the problem statement of the double-stack LPSP. In Section 4, we propose different ILP models for the LPSP and describe the solution procedure. The benefits and drawbacks of each formulation are discussed in Section 5, where we conduct extensive numerical experiments. In Section 6, we conclude the paper and outline some directions for further research.

2. Literature review

There exist several decision-making problems related to the planning and operation of container terminals (Steenken et al., 2004; Stahlbock and Voß, 2008; Carlo et al., 2014a,b). According to a classification of operational problems arising in terminals provided by Boysen et al. (2010), the LPSP comprises two out of five subproblems: deciding on the containers' positions on railcars (load planning) and on the sequence of container moves per handling equipment (load sequencing). In spite of a multitude of papers on related problems, there is to the best of our knowledge no model considering the integrated LPSP with double-stack railcars.

2.1. Load planning problem

As stated by Mantovani et al. (2018), the LPP for intermodal trains can be seen as a special case of the packing-cutting-knapsack problem (Wäscher et al., 2005). According to a common typology (Wäscher et al., 2005; Dyckhoff, 1990), the LPP is similar to a *Multiple Identical Large Object Placement Problem*: the value of weakly heterogeneous small items (standardized containers) assigned to a defined set of objects similar in size (railcars) needs to be maximized. The main difference to known packing-cutting-knapsack problems is that the objects and items are of similar dimensions (Mantovani et al., 2018).

The LPP has been extensively studied with different levels of detail. Some papers focus on weight restrictions (Bruns and Knust, 2012; Heggen et al., 2016). Others incorporate global restrictions, e.g., the unloading effort in other terminals, with the local loading rules in the considered terminal (Heggen et al., 2016; Dotoli et al., 2013, 2015, 2017; Bostel and Dejax, 1998; Cichenski et al., 2017). The variety of containers and railcars considered ranges from homogeneous containers (Bostel and Dejax, 1998; Corry and Kozan, 2006; Wang and Zhu, 2014) to a realistic variety of container and railcar characteristics (Bruns and Knust, 2012; Mantovani et al., 2018). Common objectives are the maximization of the value of loaded containers and the minimization of the costs for preparing the railcars. Other papers concentrate on the number of necessary railcars (Corry and Kozan, 2008), on the aerodynamic efficiency (Lai et al., 2008a,b) or on the wear of breaking mechanisms (Corry and Kozan, 2006). In some papers, the minimization of the handling cost is the aim of the load plan without determining the actual loading sequence of the cranes (Bostel and Dejax, 1998; Corry and Kozan, 2008). Bruns et al. (2014) focus on robust load plans considering uncertainties in the input parameters.

Double-stack trains are considered in a few studies (Corry and Kozan, 2008; Bruns and Knust, 2012; Heggen et al., 2016; Lai et al., 2008a,b; Mantovani et al., 2018), but numerical experiments are only reported in the latter three. Lai et al. (2008a,b) make simplifying assumptions that may lead to invalid load plans in practice (Mantovani et al., 2018). Pacanovsky et al. (1995) introduce a decision support tool for assigning containers to double-stack railcars and assess several loading strategies in a dynamic environment which are compared to hand-made load plans. The loading logic is based on rules, no optimization model is provided. Wang et al. (2018) propose a tabu search algorithm for a multi-objective formulation for the double-stack LPP (Lang et al., 2015). They restrict the model to the Chinese case considering only one type of railcar with one platform.

Due to the complexity of the problem, several heuristic solution methods are proposed (Bostel and Dejax, 1998; Corry and Kozan, 2008; Dotoli et al., 2015; Anghinolfi et al., 2014). The largest instances solved to optimality contain over 1,000 containers (Mantovani et al., 2018).

2.2. Load planning and sequencing problem

Most of the literature on load sequencing problems is applied to maritime container terminals (see Bierwirth and Meisel, 2010 for a thorough overview on quay crane scheduling problems and Boysen et al., 2017 for a classification scheme). Related problems are the Block Relocation Problem (BRP) and the Pre-Marshalling Problem (PMB). The BRP finds a minimal number of relocation movements for a given retrieval sequence, whereas the PMB deals with the organization of the blocks (Expósito-Izquierdo et al., 2015). For both problems, the loading sequence is an input and not subject to optimization. The load sequencing problem itself can be seen as a NP-hard asymmetric traveling salesman problem (Boysen et al., 2010).

Some sequential settings conduct the load sequencing based on a fixed load plan either by optimization (Bostel and Dejax, 1998; Wang and Zhu, 2014; Souffriau et al., 2009) or simulation (Corry and Kozan, 2008). A few studies address the integrated LPSP, but none of them permits double-stacking of containers on railcars. In addition to the dimensions discussed in the previous section, the models differ in loading and rehandling policies as well as in the number and characteristics of the handling equipment.

In all studies apart from Corry and Kozan (2006), the scope of the problem is limited to the loading process assuming that the railcars have been unloaded before. Moreover, the vast majority of studies consider one handling equipment at a time. This relates to the yard partition problem, which divides intermodal terminals into disjunct areas levelling the workload for cranes (Boysen and Fliedner, 2010; Boysen et al., 2010). However, a few works involve more than one crane (e.g., Ambrosino et al., 2016; Otto et al., 2017).

Some settings restrict the loading sequence of the train from its head to its rear (Ambrosino et al., 2011; Ambrosino and Caballini, 2018). A few studies investigate the impact of forbidding non-sequential loading orders and find that the complexity of the problem is considerably reduced (Ambrosino et al., 2013; Ambrosino and Siri, 2014). As the instances with non-sequential loading policy could not be solved to optimality, no consequences on the number of rehandlings are reported. Others impose a non-sequential loading of the railcars (Ambrosino and Siri, 2015; Ambrosino et al., 2016; Corry and Kozan, 2006). All related studies allow rehandlings of containers, yet the processes differ depending on the considered setting. In Corry and Kozan (2006), a container is rehandled if it cannot be directly transferred from an inbound truck to an outbound train. In the other works, rehandlings occur if a needed container cannot be accessed in the storage area and other blocking containers must be retrieved first. Some computational studies investigate the consequences of forbidding rehandlings (Ambrosino et al., 2013; Ambrosino and Siri, 2014). Contrary to the prohibition of non-sequential loading orders, the complexity of the problem remains high and it cannot be quickly solved with a general-purpose solver (Ambrosino and Siri, 2014).

Corry and Kozan (2006) treat a dynamic setting with uncertainty in the data. They adapt the load plan in a rolling horizon environment by solving a deterministic model with updated data each time an event is triggered.

Typical objectives of the LPSP are the minimization of the handling cost consisting of rehandlings (Corry and Kozan, 2006; Ambrosino et al., 2011, 2013, 2016; Ambrosino and Siri, 2014; Ambrosino and Caballini, 2018) and costs for the distance covered by the handling equipment (Corry and Kozan, 2006; Ambrosino et al., 2013; Ambrosino and Siri, 2014). The latter costs are interpreted in different ways. Corry and Kozan (2006) only take into account the costs if the slot assignment of a container is changed compared to a prior load plan, because they assume that trucks deliver containers straight to the initially assigned railcar. Ambrosino et al. (2013) consider the distance travelled by a GC along the track, whereas Ambrosino and Siri (2014) only consider unproductive backward movements of the crane. The latter two studies exclude the costs for the distance of RSs that place the containers next to the assigned railcar. Most works additionally incorporate objectives related to the LPP discussed in Section 2.1.

Due to the complexity of the problem, exact algorithms are applied only to small instances. Some studies develop tailored heuristic solution techniques (Ambrosino et al., 2011; Ambrosino and Siri, 2015; Ambrosino and Caballini, 2018). The largest instances solved to optimality are small compared to the LPP. Ambrosino et al. (2013) apply a generic all-purpose solver on instances comprising 40 containers allowing a non-sequential loading order and rehandling of containers. Ambrosino and Siri (2015) propose three solution procedures for the LPSP: two make use of a MIP solver provided with an initial solution either by a tailored constructive method or by an optimal solution of a simplified model imposing a strict loading order, the third is a heuristic solution technique. Even the smallest instances (40 containers) cannot be optimally solved by any of the solution techniques. Ambrosino et al. (2016) consider two cranes and solve instances with 24 containers to optimality with a MIP solver. Recall that the former models do not consider double-stacking of containers. None of the discussed papers provides a tailored exact solution technique.

2.3. Summary

As this literature review shows, to the best of our knowledge there is no model that treats the LPSP for double-stack intermodal trains. So far, Mantovani et al. (2018) present the only formulation that takes a high variety of loading patterns dealing with double-stack railcars into account. However, this model ignores the sequencing part, i.e., the sequencing is done in a later stage. Since the load planning does not take into account the sequencing constraints – namely the position of the containers in the stacks and the characteristics of the handling equipment – unproductive movements can be inevitably part of the loading process even though they could likely be avoided by an integrated planning and sequencing methodology. All models for the LPSP consider single-stack railcars. Additionally, they either lack in realistic variety of the containers (Corry and Kozan, 2006) and in flexibility of the loading sequence (Ambrosino et al., 2011; Ambrosino and Caballini, 2018), or they exclude overbooking (Corry and Kozan, 2006).

3. The load planning and sequencing problem for double-stack intermodal trains

The LPSP is an integrated operational problem governed by the characteristics of the layout of the terminal, the containers, the railcars, and the handling equipment. In the following, we use common terminology in the rail industry. It may differ from the vocabulary used, e.g., in ports. The LPSP is solved for each block that has a given destination and departure time. We assume that the LPSP is solved for one given handling equipment. If a handling equipment simultaneously loads multiple blocks, the same methodology applies by considering the union of the sets of containers and railcars for the different blocks.

3.1. Intermodal rail-road terminals

An intermodal rail-road terminal consists of several areas. This paper considers a terminal layout that is inspired by a real terminal of our industrial partner (see Fig. 1) where RSs are used to move containers. Trucks arrive at the terminal to unload their containers $i \in N$. The handling equipment stores the containers in the storage area. Direct rail-road or rail-rail transfers are not considered in this paper. Cranes and vehicles may move in the gray zones (Figure 1), but terminal-specific restrictions may apply.

Containers are placed on top of each other up to a given maximum height in *stacks*. Multiple stacks are lengthwise contiguously grouped in so-called *lots* which are located along the track. Containers are divided into lots according to their block (i.e., destination and train departure) and size (20 ft containers are stacked separately). The coordinates X, Y and Z indicate the exact position of each container in the storage area. The X-coordinate numbers the lot. The Y-coordinate indicates the stack and the Z-coordinate specifies the vertical position of a container. Depending on the handling equipment, the storage area can

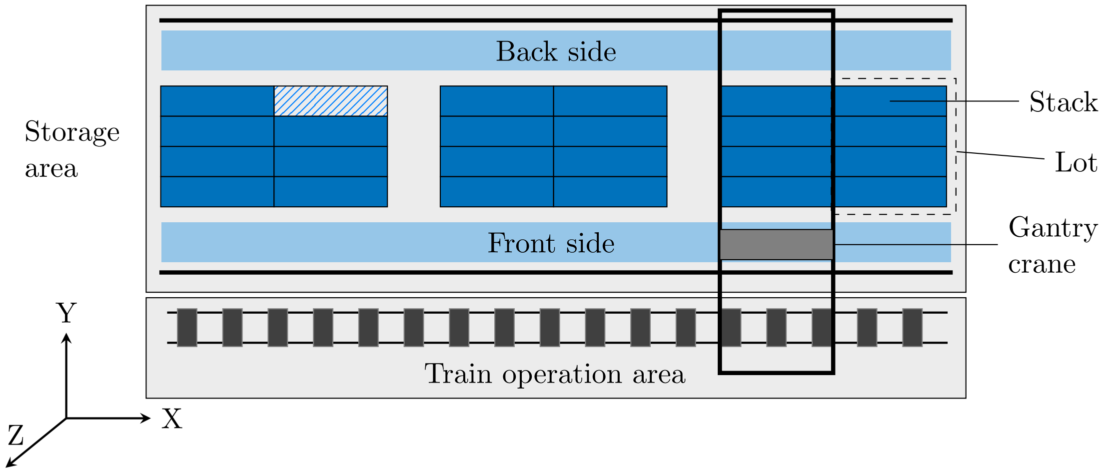

Figure 1: Aerial view of the considered container terminal layout

be accessed from above, from the front side of the stack (seen from the track), or from its back (as it is appropriate for the lined container in Figure 1). For the sake of comparability of the computational experiments, we use the same layout for both types of handling equipment.

For loading a container, the handling equipment retrieves it from the storage area, carries it to a railcar, and places it onto its assigned slot $q \in Q$ where Q is the set of all slots on railcars in a given block. If the container is not directly accessible, it is necessary to rehandle the blocking containers. We assume that the railcars have already been unloaded, the corresponding decision-making problem is out of the scope of this paper.

3.2. Containers

A container $i \in N$ can be of any of the standardized types and N is the set of outbound containers for a given block. We note that this set can exceed the available capacity (i.e., we consider overbooking). Each container is characterized by its length l_i (in the North American market sizes $h \in H$ are 20 ft, 40 ft, 45 ft, 48 ft and 53 ft, we denote the related subsets N_h), its height (low-cube or high-cube containers in the sets N^{LC} and N^{HC}, resp.), its weight g_i, and its type (e.g., refrigerated, tank). Depending on the container type, different technical loading restrictions may apply and depending on the content, the customer, and the due time, each container has a specific commercial value π_i. This can be an actual commercial value, or represent a loading priority (e.g., reflecting a due date or special service). Last, we define the constant m_i that takes value 0.5 for containers of length 20 ft and 1 for all other containers.

3.3. Intermodal double-stack railcars

A block consists of a sequence of railcars $j \in J$ where each railcar is defined by its type listed in a catalogue used in practice (Association of American Railroads, 2017). A railcar j has between one and five platforms $p \in P_j$. Platforms $p \in P$ can be either single-stack or double-stack and are characterized by various technical features such as the length of the bottom slot L_p, a maximum weight-carrying capacity G_p, and a set of slots Q_p. We denote bottom slots with the parameter $\mu_q = 1$ and define the set of all bottom slots as Q_μ. The parking position of slot q on the track is referred to as X^q.

Railcars can be used in several configurations differing in the number and length of loaded containers (Bruns and Knust, 2012). We refer to these configurations as loading patterns

$k \in K$. Since loaded containers can influence the feasible set of container combinations on neighboring platforms of the same railcar, the loading patterns are derived by railcar and not by platform (see Association of American Railroads, 2017 and Mantovani et al., 2018 for further explanations). The set of feasible loading patterns for railcar j is denoted K_j. For each pattern k, we define the number of containers of length h to be put on platform p as $n^h_{k(p)}$. The setup costs for changing the configuration of railcar j are denoted τ_j.

Besides the loading patterns, additional constraints related to the weights of the containers apply. For example, to meet the regulations related to a maximum height of the center of mass for each loaded platform, a parameter for a maximum weight for top containers is derived per bottom container and platform: $c_i^{LC_p}$ and $c_i^{HC_p}$ for low-cube and high-cube containers, respectively (Association of American Railroads, 2017; Mantovani et al., 2018). The weight of each individual container is important in order to make the best use of the available capacity while satisfying this constraint.

Finally, 20 ft containers can solely be assigned to bottom slots and a top slot can only be loaded if either two 20 ft containers or one container measuring at least 40 ft have been loaded before in the bottom slot of the platform. No restrictions are given on the loading order of the block, i.e., we consider non-sequential loading.

3.4. Handling equipment

Terminals are equipped with special handling equipment that handle the containers within the terminal. We consider GCs and RSs (see examples of both in Figure 2), because both can be present in intermodal rail terminals and both are considered in the literature. GCs are immobile facilities that pick a container from above. RSs, by contrast, are vehicles that lift containers from the side. As previously mentioned, the container layout we consider is inspired from a real terminal that uses RSs only. The focus of this paper is not the optimization of the terminal layout; we assume the same layout for both GCs and RSs. However, it is only realistic for the latter. We note that in large terminals, in particular ports, multiple handling equipment may be used to load high demand blocks.

(a) A gantry crane (Adobe Stock, 2020a)   (b) A reach stacker (Adobe Stock, 2020b)

Figure 2: Examples of handling equipment

As not every container can be retrieved by the handling equipment at any given loading stage $t \in T$, we define accessibility rules for each container. For a GC the rules are simple: a container can only be retrieved if it is the uppermost container of its stack.

For RSs, however, the rules are more complex. RSs retrieve containers either from the front side or from the back side of the lot (e.g., for the lined container in Figure 1). For the latter, the covered distance increases and therefore, we denote this movement as *detour*. As all container positions and the possible crane movements are given, impossible sequences between two containers can be derived a priori. We therefore define pairs of containers (i, i'), such that container i' must be taken out of the stack before container i. In other words, the loading sequence i before i' is forbidden.

We define three sets of forbidden loading sequences: one for the GC (M^G), one for the RS retrieving the containers from the front (M^{RF}), and one for the RS from the back (M^{RB}). The rules are derived by geometric dependencies and technical limitations of the handling equipment. We define the container pairs (i, i') using the following rules (cf. Figure 3):

(a) container i is below container i': $X^i = X^{i'}$, $Y^i = Y^{i'}$, $Z^i < Z^{i'}$

(b) container i is hidden by i': $X^i = X^{i'}$, $Y^{i'} < Y^i$, $Z^i \leq Z^{i'}$

(c) container i is more than three positions behind i': $X^i = X^{i'}$, $Y^{i'} + 3 \leq Y^i$

(d) the mass of container i (g_i) exceeds the threshold θ^1 for being lifted over one container stack: $X^i = X^{i'}$, $Y^{i'} + 1 = Y^i$, $g_i > \theta^1$

(e) the mass of container i (g_i) exceeds the threshold θ^2 for being lifted over two container stacks: $X^i = X^{i'}$, $Y^{i'} + 2 = Y^i$, $g_i > \theta^2$

(f) – (i) analogously to (b) - (e) restrictions for RSs retrieving a container from the back side. The rules are rotated with respect to the Y-coordinate.

The set M^G comprises all container pairs fitting the rule (a). The set M^{RF} relates to the rules (a) – (e), and for the set M^{RB}, the rules (a), and (f) – (i) are relevant. These forbidden sequences are illustrated in Figure 3 and can easily be adapted to local restrictions.

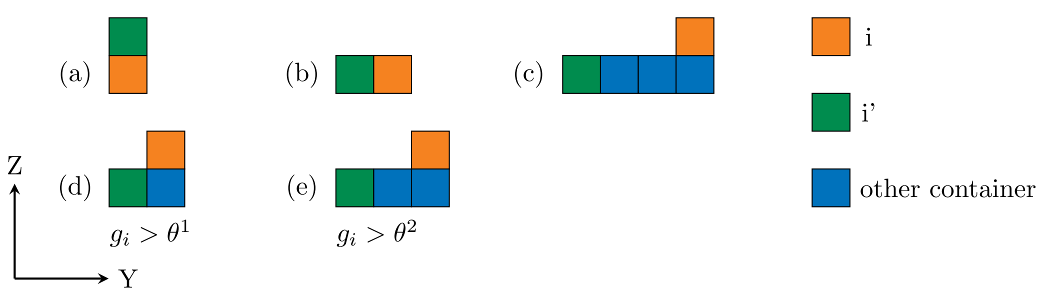

Figure 3: Overview of forbidden loading sequence $i \rightarrow i'$ (for GCs: (a), for RSs from the left hand side (a) – (e))

If a forbidden sequence is executed, a container is inaccessible. Then, either a detour (if possible) or a *rehandle* is necessary. Rehandles are unproductive movements that are necessary to take out all blocking containers, such that the required container can be reached (Ambrosino and Caballini, 2018). The blocking containers may stay in the terminal or be loaded to the train at a later moment and must then be touched again.

3.5. Challenges and objective

The LPSP minimizes the cost for not loaded containers, the setup costs of the railcars, and the handling costs. The handling costs comprise the distance covered by the handling equipment, the number of rehandled containers, and the number of detours. As illustrated in Figure 1, the handling equipment covers a distance expressed by taxicab geometry which is used for both types of handling equipment in the objective function. In this context, β^1_{iq} represents the distance cost between the stacking position of container i and slot q, β^2 is the cost for rehandling a container, and in the case of a RS, β^3 represents the cost for a detour.

The considered problem comprises numerous interdependent decisions. The decision on the loading pattern defines how many containers of each size can be loaded on each platform. The assignment of containers to slots is made by respecting the loading patterns and additional constraints such as weight restrictions. In particular, constraints on the center of mass depend on individual container weights which makes it inappropriate to simplify the problem by aggregating containers into types. This assignment imposes restrictions on feasible loading sequences as containers assigned to top slots can only be loaded once the corresponding bottom slots are filled.

Summarizing, the LPSP is defined as follows: Given a set of containers stored in a terminal with their characteristics and position, a set of railcars, a handling equipment, and the relevant constraints, determine the subset of containers to load, the exact way and sequence of retrieving and loading them, such that the value of not loaded containers and the handling cost is minimized.

4. ILP formulations and solution procedure

In this section, we introduce several formulations for the LPSP with double-stack railcars. All formulations are based on the ILP for the LPP proposed by Mantovani et al. (2018) (Sec. 4.1). In Section 4.2, we introduce sets, notation, and decision variables that are used in all formulations. Next, in Sections 4.3 – 4.6, we define four different ILP formulations (denoted A1, A2, B1, and B2). They differ in the number of constraints and variables, as well as in the strength of their LP relaxation. Since there is no clear dominance relationship between the formulations, we report all four formulations along with their computational results. The formulations can most easily be distinguished by the meaning of the variables (cf. Table 1 for an overview).

Variables	z_{iqt}		z_{it}, b_{qt}	
Process-oriented	A1	Sec. 4.3	B1	Sec. 4.5
State-oriented	A2	Sec. 4.4	B2	Sec. 4.6

Table 1: Names of the LPSP formulations and references to the subsections where they are delineated

We start by explaining the difference between the formulations 1 and 2. The sequencing variables in the formulations A1, and B1 are process-oriented: they indicate the stage at which a container loading occurs (loaded *at* stage t). For example, if a container is loaded at stage 1, the value of the sequencing variable for stage 2 is 0. By contrast, the variables of the remaining formulations are state-oriented: they indicate if a given container has been loaded at or before a certain stage (loaded *by* stage t). Taking the same example, the sequencing variable for stage 2 takes value 1 in these formulations. The latter simplifies the writing of some constraints at the cost of additional constraints. The distinction between both types of sequencing variables can be found in the literature in another context. Namely, Expósito-Izquierdo et al. (2015) delineate a model for the BRP that includes both process- and state-oriented variables.

We now turn our attention to the difference between the A and B formulations: The A formulations contain three-index sequencing variables resulting in a high number of variables and a rather low number of constraints. These variables tie together a container, a slot, and a stage (z_{iqt}). The B formulations use two-index sequencing variables which carry less information. These variables only link a container to a stage. In addition, the B formulations make use of a set of variables related to the loading state of each bottom slot. These extra variables join a slot to a stage (z_{it} and b_{qt}). This results in fewer variables but more constraints.

Some variants of the formulations with a simplified cost structure are presented in Section 4.7. In Section 4.8, we describe the proposed solution procedure.

4.1. Formulation of the load planning problem

A solution to the LPP consists of, for a given block, the assignment of a subset of containers to slots on a given set of railcars. The railcars in the North American fleet exhibit a high variety in loading patterns that are crucial to take into account in order to achieve accurate and feasible load planning solutions. As the methodology proposed by Mantovani et al. (2018) is the only one that addresses this situation, we use it as a basis for the LPSP. In this section, we describe the corresponding ILP formulation. An overview of the parameters and variables is given in Table 2.

The LPP presented here differs in one set of constraints (3) from the original model. In Mantovani et al. (2018), the assignment of a container to a slot on a given platform can be determined or changed in a post-processing step. Since we define the load planning and sequencing simultaneously, we add constraints (3) to ensure a feasible slot assignment.

Let us start by introducing the four sets of binary variables that are used in the model: The decision variables w_{jk} assign loading patterns to railcars and v_{iq} take care of the container-slot assignments. The auxiliary variables y_{ip} and x_{ij} assign containers to platforms and to railcars, respectively. The formulation of the LPP is:

$$\min \sum_{i \in N} \pi_i \left(1 - \sum_{q \in Q} v_{iq}\right) + \sum_{j \in J} \tau_j \sum_{k \in K_j} w_{jk} \tag{1}$$

$$\text{s.t.} \quad \sum_{q \in Q} v_{iq} \leq 1 \qquad \forall i \in N \tag{2}$$

$$\sum_{i \in N} m_i v_{iq} \leq 1 \qquad \forall q \in Q \tag{3}$$

$$y_{ip} = \sum_{q \in Q_p} v_{iq} \qquad \forall i \in N, \forall p \in P \tag{4}$$

$$x_{ij} = \sum_{p \in P_j} y_{ip} \qquad \forall i \in N, \forall j \in J \tag{5}$$

$$\sum_{k \in K_j} w_{jk} \leq 1 \qquad \forall j \in J \tag{6}$$

$$\sum_{k \in K_j} n^h_{k(p)} w_{jk} = \sum_{i \in N_h} y_{ip} \qquad \forall p \in P_j, \forall j \in J, \forall h \in H \tag{7}$$

Sets		N	all containers
H	container lengths	N_h	containers of length h
J	railcars	N^{HC}	high-cube containers
K	loading patterns	N^{LC}	low-cube containers
K_j	loading patterns of railcar j	P	all platforms
M^G	forbidden loading sequences for GCs	P_j	platforms of railcar j
M^{RF}	forbidden loading sequences for RSs from the front	Q	all slots
		Q_p	slots of platform p
M^{RB}	forbidden loading sequences for RSs from the back	Q_μ	bottom slots
		T	stages
Indices		k	loading pattern with $k \in K$
h	container length with $h \in H$	p	platform with $p \in P$
i	container with $i \in N$	q	slot with $q \in Q$
j	railcar with $j \in J$	t	stage with $t \in T$
Parameters		t^f / t^l	first stage / last stage
$c_i^{LC_p}$	maximum weight of a low-cube container to be put onto container i on platform p	X^i	X coordinate of container i in storage area
		X^q	X coordinate of slot q
$c_i^{HC_p}$	maximum weight of a high-cube container to be put onto container i on platform p	Y^i	Y coordinate of container i in storage area
		Z^i	Z coordinate of container i in storage area
g_i	weight of container i	β^1_{iq}	distance cost between container i and slot q
G_p	maximum weight-carrying capacity of platform p	β^2	cost for rehandling a container
		β^3	cost for a detour
l_i	length of container i	θ^1 / θ^2	maximum weight of containers being lifted by a RS over one / two container stack(s)
L_p	length of the bottom slot of platform p		
m_i	takes value 0.5 if length l_i is 20 ft, 1 else	μ_q	takes value 1 if slot q is a bottom slot, 0 else
$n^h_{k(p)}$	number of containers of length h to be put on platform p in pattern k	π_i	commercial value of container i
		τ_j	setup cost of railcar j
Binary variables		y_{ip}	$= 1$ if container i is assigned to platform p, 0 else
d_i	$= 1$ if container i is rehandled, 0 else		
u_{qi}	$= 1$ if handling equipment reverses from slot q to container i, 0 else	w_{jk}	$= 1$ if loading pattern k is assigned to railcar j, 0 else
v_{iq}	$= 1$ if container i is assigned to slot q, 0 else	γ_i	$= 1$ if container i is retrieved from the back, 0 else
x_{ij}	$= 1$ if container i is assigned to railcar j, 0 else		

Table 2: Notation of sets, indices, parameters, and variables

$$\sum_{i \in N} \sum_{q \in Q_p} \mu_q v_{iq} l_i \leq L_p \qquad \forall p \in P \tag{8}$$

$$\sum_{i \in N} y_{ip} g_i \leq G_p \qquad \forall p \in P \tag{9}$$

$$\sum_{i \in N^{LC}} \sum_{q \in Q_p} (1 - \mu_q) v_{iq} g_i \leq \sum_{i \in N} \sum_{q \in Q_p} \mu_q v_{iq} c_i^{LCp} \qquad \forall p \in P \tag{10}$$

$$\sum_{i \in N^{HC}} \sum_{q \in Q_p} (1 - \mu_q) v_{iq} g_i \leq \sum_{i \in N} \sum_{q \in Q_p} \mu_q v_{iq} c_i^{HCp} \qquad \forall p \in P \tag{11}$$

$$w_{jk} \in \{0, 1\} \qquad \forall j \in J, \forall k \in K \tag{12}$$

$$v_{iq} \in \{0, 1\} \qquad \forall i \in N, \forall q \in Q \tag{13}$$

$$y_{ip} \in \{0, 1\} \qquad \forall i \in N, \forall p \in P \tag{14}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in N, \forall j \in J. \tag{15}$$

The objective (1) is to minimize the weighted costs for containers left in the terminal and the setup costs for each loaded railcar. Constraints (2) make sure that each container is assigned to only one slot, whereas (3) ensure that at most one or two containers are assigned to each slot. Constraints (4) and (5) link the platform variables y_{ip} with the slot variables v_{iq} and the railcar variables x_{ij}, respectively. Constraints (6) limit the number of chosen patterns per railcar to 1 (empty railcars have no assigned pattern) and constraints (7) link the attributes of the pattern to the loading variables of each platform. Constraints (8) and (9) guarantee a feasible loading with respect to the maximum length and weight of each platform. The center of gravity constraints are formulated in (10) for low-cube containers and in (11) for high-cube containers, respectively. Finally, we note that Mantovani et al. (2018) additionally introduce six types of technical restrictions. As these restrictions do not affect the load sequencing, we omit them in this paper. We refer to Mantovani et al. (2018) for a more in-depth explanation of the model as well as for an extensive numerical study.

4.2. Notation and variables common to all of the LPSP formulations

We assume one container loading per stage $t \in T$, where we refer to the first stage as t^f and to the last one as t^l. We ensure that all loadings are contiguous at the beginning of the time horizon. We define three sets of binary variables: d_i takes value 1 if container i is rehandled and u_{qi} takes value 1 if the handling equipment reverses from slot q to the stacking position of container i. In the case of a RS an additional variable γ_i takes value 1 if it makes a detour for container i. These definitions lead to the following constraints:

$$d_i \in \{0,\, 1\} \qquad \forall i \in N \tag{16}$$

$$\gamma_i \in \{0,\, 1\} \qquad \forall i \in N \tag{17}$$

$$u_{qi} \in \{0,\, 1\} \qquad \forall q \in Q,\, \forall i \in N. \tag{18}$$

As discussed in Section 3.4, we consider two types of handling equipment: a RS and a GC. Nevertheless, we choose to present the formulations in Sections 4.3 to 4.6 for RS only. There are several reasons for this choice. First, the RS is our main scenario as the real terminal layout we consider has been designed for RS movements. Second, it is easy to transform the formulation to one corresponding to a GC. In this case, the set of M^{RF} is replaced by M^G and the γ_i variables are left out both in the objective function and in the constraints. Moreover, all constraints defined over M^{RB} can be dropped.

4.3. Formulation A1

In the formulations A1 and A2, we consider three-index sequencing variables z_{iqt}. These formulations require a high number of variables but come with a rather low number of constraints. We define z_{iqt} as a binary decision variable taking value 1 if and only if container i is loaded in slot q in stage t. The LPSP can be written as follows:

$$\min \sum_{i\in N} \pi_i \left(1 - \sum_{q\in Q} v_{iq}\right) + \sum_{j\in J} \tau_j \sum_{k\in K_j} w_{jk} + \sum_{i\in N} \left(\sum_{q\in Q} \beta^1_{iq}\left(v_{iq} + u_{qi}\right) + \beta^2 d_i + \beta^3 \gamma_i\right) \tag{19}$$

s.t. Constraints (2) – (18)

$$\sum_{t\in T} z_{iqt} = v_{iq} \qquad \forall i \in N,\, \forall q \in Q \tag{20}$$

$$\sum_{i\in N}\sum_{q\in Q} z_{iqt} \leq 1 \qquad \forall t \in T \tag{21}$$

$$\sum_{i\in N}\sum_{q\in Q} z_{iq,t+1} \leq \sum_{i\in N}\sum_{q\in Q} z_{iqt} \qquad \forall t \in T \setminus \{t^l\} \tag{22}$$

$$z_{iqt} + v_{iq} + \sum_{s\in Q} z_{i's,t+1} - u_{qi'} \leq 2 \qquad \begin{array}{l}\forall i \in N,\, \forall i' \in N,\\ \forall q \in Q,\, \forall t \in T \setminus \{t^l\}\end{array} \tag{23}$$

$$\sum_{q\in Q}\sum_{\ell=0}^{t} z_{iq\ell} \leq \sum_{q\in Q}\sum_{\ell=0}^{t} z_{i'q\ell} + d_{i'} + \gamma_i \qquad \forall (i,\, i') \in M^{RF},\, \forall t \in T \tag{24}$$

$$\sum_{q\in Q}\sum_{\ell=0}^{t} z_{iq\ell} \leq \sum_{q\in Q}\sum_{\ell=0}^{t} z_{i'q\ell} + d_{i'} - \gamma_i + 1 \qquad \forall (i,\, i') \in M^{RB},\, \forall t \in T \tag{25}$$

$$\sum_{i\in N}\sum_{q\in Q_p}\sum_{\ell=0}^{t} z_{iq\ell}\,(1-\mu_q) \leq \sum_{i\in N}\sum_{q\in Q_p}\sum_{\ell=0}^{t} m_i z_{iq\ell}\mu_q \qquad \forall p \in P,\, \forall t \in T \tag{26}$$

$$z_{iqt} \in \{0,\, 1\} \qquad \forall i \in N,\, \forall q \in Q,\, \forall t \in T. \tag{27}$$

Each assigned container must be loaded in its assigned slot (20). The number of loadings per stage is limited to 1 (21). All loadings must be contiguous, i.e., no stage without loading is allowed between two stages with loadings (22). Constraints (23) link the assignment variables v_{iq}, the sequencing variables z_{iqt} and the reverse variables u_{qi}: if container i is loaded in slot q at stage t and container i' is loaded in any slot s at the following stage $t+1$, the reverse variable $u_{qi'}$ must take value 1. The accessibility of the containers must be respected both for containers loaded from the front side (24) and from the back side (25) of the lot. The forbidden sequence is bypassed if i' is rehandled. The constraints (24) are always true if i is retrieved by a detour ($\gamma_i = 1$) and for (25), the reverse holds true ($\gamma_i = 0$). Constraints (26) ensure that the top slot of each platform is loaded only if the bottom slot has been filled. The sum of the left-hand sequencing variables for the top slot may only take value 1 if the sum of the right-hand variables equals 1, i.e., either a container measuring at least 40 ft or two 20 ft containers have been loaded at a prior stage.

4.4. Formulation A2

In the sets of constraints (24) – (26), we sum over many stages to obtain the information whether container i has been loaded in slot q. In formulation A2, we redefine z_{iqt} as a state-oriented variable to decide whether container i is loaded in slot q by stage t. Once container i has been loaded in slot q, the value of the associated z_{iqt} variables is 1 for all following stages. The model then becomes:

$$\min \sum_{i\in N} \pi_i \left(1 - \sum_{q\in Q} v_{iq}\right) + \sum_{j\in J} \tau_j \sum_{k\in K_j} w_{jk} + \sum_{i\in N}\left(\sum_{q\in Q} \beta^1_{iq}\,(v_{iq} + u_{qi}) + \beta^2 d_i + \beta^3 \gamma_i\right) \tag{28}$$

s.t. Constraints (2) – (18) and (27)

$$z_{iqt^l} = v_{iq} \qquad \forall i \in N,\, \forall q \in Q \tag{29}$$

$$z_{iqt} \leq z_{iq,t+1} \qquad \begin{array}{l}\forall i \in N,\, \forall q \in Q,\\ \forall t \in T \setminus \{t^l\}\end{array} \tag{30}$$

$$\sum_{i\in N}\sum_{q\in Q} z_{iqt} \leq \sum_{i\in N}\sum_{q\in Q} z_{iq,t-1} + 1 \qquad \forall t \in T \setminus \{t^f\} \tag{31}$$

$$\sum_{i\in N}\sum_{q\in Q} (z_{iq,t+1} - z_{iqt}) \leq \sum_{i\in N}\sum_{q\in Q} (z_{iqt} - z_{iq,t-1}) \qquad \forall t \in T \setminus \{t^f, t^l\} \tag{32}$$

$$z_{iqt} - z_{iq,t-1} + \sum_{s\in Q} \left(z_{i's,t+1} - z_{i'st}\right) + v_{iq} - u_{qi'} \leq 2 \qquad \begin{array}{l}\forall i \in N, \forall i' \in N, \forall q \in Q,\\ \forall t \in T \setminus \{t^f, t^l\}\end{array} \tag{33}$$

$$\sum_{q\in Q} z_{iqt} \leq \sum_{q\in Q} z_{i'qt} + d_{i'} + \gamma_i \qquad \forall (i,\, i') \in M^{RF}, \forall t \in T \tag{34}$$

$$\sum_{q\in Q} z_{iqt} \leq \sum_{q\in Q} z_{i'qt} + d_{i'} - \gamma_i + 1 \qquad \forall (i,\, i') \in M^{RB}, \forall t \in T \tag{35}$$

$$\sum_{i\in N}\sum_{q\in Q_p} z_{iqt}\,(1 - \mu_q) \leq \sum_{i\in N}\sum_{q\in Q_p} m_i z_{iqt} \mu_q \qquad \forall p \in P, \forall t \in T. \tag{36}$$

Constraints (29) make sure that each assigned container is loaded by the end of the time horizon t^l. Each z_{iqt} variable takes value 1 once a container has been loaded (30). Constraints (31) limit the number of loadings per stage to one. All stages without loadings are contiguous at the end (32). Constraints (33) make sure that the variables u_{qi} for the reverse movements of the handling equipment are correctly set. The accessibility of containers must be respected both for containers loaded from the front (34) and from the back (35). Constraints (36) ensure that the top slot is only loaded after the bottom slot has been filled.

Comparing the constraints of A1 and A2, those of A2 ensuring a correct loading (34) – (36) are simplified at the cost of a larger number of additional constraints (30).

4.5. Formulation B1

As the A formulations involve a large number of sequencing variables ($|N||Q||T|$), we now present a formulation requiring fewer decision variables. We define binary variables z_{it} taking value 1 if container i is loaded on the railcar in stage t. The slot to which a container is assigned can be obtained from the v_{iq} variables (13) of the LPP. This leads to a reduction in the number of z_{it} variables by a factor $|Q|$. In order to simplify the constraints related to the correct loading sequence of containers to double-stack railcars, we introduce the auxiliary variable b_{qt}, which takes value 1 if bottom slot q is fully loaded at stage t, and 0 otherwise. The sequencing variables take value 1 if container i is loaded at stage t, and 0 otherwise. We formulate the problem as follows:

$$\min \sum_{i\in N} \pi_i \left(1 - \sum_{q\in Q} v_{iq}\right) + \sum_{j\in J} \tau_j \sum_{k\in K_j} w_{jk} + \sum_{i\in N} \left(\sum_{q\in Q} \beta^1_{iq}\,(v_{iq} + u_{qi}) + \beta^2 d_i + \beta^3 \gamma_i\right) \tag{37}$$

s.t $\qquad$ Constraints (2) – (18)

$$\sum_{t\in T} z_{it} = \sum_{q\in Q} v_{iq} \qquad \forall i \in N \tag{38}$$

$$\sum_{i\in N} z_{it} \leq 1 \qquad \forall t \in T \tag{39}$$

$$\sum_{i\in N} (z_{i,t+1} - z_{it}) \leq \sum_{i\in N} (z_{it} - z_{i,t-1}) \qquad \forall t \in T \setminus \{t^f, t^l\} \tag{40}$$

$$z_{it} + v_{iq} + z_{i',t+1} - u_{qi'} \leq 2 \qquad \forall i \in N, \forall i' \in N, \forall q \in Q, \forall t \in T \setminus \{t^l\} \tag{41}$$

$$\sum_{\ell=0}^{t} z_{i\ell} \leq \sum_{\ell=0}^{t} z_{i'\ell} + d_{i'} + \gamma_i \qquad \forall (i,\, i') \in M^{RF}, \forall t \in T \tag{42}$$

$$\sum_{\ell=0}^{t} z_{i\ell} \leq \sum_{\ell=0}^{t} z_{i'\ell} + d_{i'} - \gamma_i + 1 \qquad \forall (i,\, i') \in M^{RB}, \forall t \in T \tag{43}$$

$$\sum_{\ell=0}^{t} z_{i\ell} + \sum_{q \in Q_p} v_{iq} \left(1 - \mu_q\right) \leq \sum_{q \in Q_p \cap Q_\mu} \sum_{\ell=0}^{t} b_{q\ell} + 1 \qquad \forall p \in P, \forall i \in N, \forall t \in T \tag{44}$$

$$b_{qt} \leq \sum_{\ell=0}^{t} z_{i\ell} - v_{iq} + 1 \qquad \forall i \in N, \forall q \in Q_\mu, \forall t \in T \tag{45}$$

$$z_{it} \in \{0,\, 1\} \qquad \forall i \in N, \forall t \in T \tag{46}$$

$$b_{qt} \in \{0,\, 1\} \qquad \forall q \in Q_\mu, \forall t \in T. \tag{47}$$

Constraints (38) make sure that each assigned container is loaded during the time horizon, whereas (39) limit the number of loadings per stage to 1. All stages without loading are shifted to the end (40). Constraints (41) ensure that the variables u_{qi} take value 1 if the handling equipment reverses from slot q to container i. Constraints (42) ensure that either i' is moved before i, i' is rehandled or i is picked from the back side. Constraints (43) work equivalently for the containers that are retrieved from the back side. Constraints (44) manage the correct loading sequence of each platform: i can only be loaded in the top slot in stage t if the bottom slot of the same platform has been filled before. The additional constraints (45) ensure the synchronization between the z_{it} and the b_{qt} variables. Remark that if two 20 ft containers are assigned to the bottom slot q, b_{qt} can only take value 1 after the second loading.

4.6. Formulation B2

Similar to the formulation A2 (Section 4.4), this formulation aims at simplifying the constraints (42) - (44) and therefore defines the decision variable z_{it} to to be 1 if container i has been loaded by stage t. Analogously to the formulation B1, the auxiliary variable b_{qt} is introduced and takes value 1 if slot q has been (fully) loaded by stage t. These definitions lead to the following model:

$$\min \sum_{i \in N} \pi_i \left(1 - \sum_{q \in Q} v_{iq}\right) + \sum_{j \in J} \tau_j \sum_{k \in K_j} w_{jk} + \sum_{i \in N} \left(\sum_{q \in Q} \beta^1_{iq} \left(v_{iq} + u_{qi}\right) + \beta^2 d_i + \beta^3 \gamma_i\right) \tag{48}$$

s.t $\qquad$ Constraints (2) – (18) and (46) – (47)

$$z_{it^l} = \sum_{q \in Q} v_{iq} \qquad \forall i \in N \tag{49}$$

$$z_{it} \leq z_{i,t+1} \qquad \forall i \in N, \forall t \in T \setminus \{t^l\} \tag{50}$$

$$\sum_{i \in N} z_{it} \leq \sum_{i \in N} z_{i,t-1} + 1 \qquad \forall t \in T \setminus \{t^f\} \tag{51}$$

$$\sum_{i \in N} \left(z_{i,t+1} - z_{it}\right) \leq \sum_{i \in N} \left(z_{it} - z_{i,t-1}\right) \qquad \forall t \in T \setminus \{t^f, t^l\} \tag{52}$$

$$z_{it} - z_{i,t-1} - z_{i't} + z_{i',t+1} + v_{iq} - u_{qi'} \leq 2 \qquad \forall i \in N, \forall i' \in N, \forall q \in Q, \forall t \in T \setminus \{t^f, t^l\} \tag{53}$$

$$z_{it} \leq z_{i't} + d_{i'} + \gamma_i \qquad \forall (i,\, i') \in M^{RF}, \forall t \in T \tag{54}$$

$$z_{it} \leq z_{i't} + d_{i'} - \gamma_i + 1 \qquad \forall (i,\, i') \in M^{RB}, \forall t \in T \tag{55}$$

$$z_{it} \leq \sum_{q \in Q_p} \left(\mu_q b_{qt} - v_{iq}\left(1 - \mu_q\right)\right) + 1 \qquad \forall p \in P, \forall i \in N, \forall t \in T \tag{56}$$

$$b_{qt} \leq z_{it} - v_{iq} + 1 \qquad \forall i \in N, \forall q \in Q_\mu, \forall t \in T. \tag{57}$$

Constraints (49) ensure that each assigned container has been loaded in the last stage. Constraints (50) ensure that variables z_{it} stay at value 1 once i has been loaded, whereas constraints (51) limit the number of loadings per stage to 1. Constraints (52) forbid stages without loading between two with loadings. Constraints (53) make sure that the variables v_{iq}, z_{it}, and u_{qi} are correctly synchronized. Constraints (54) and (55) take care of the accessibility of the containers in the stack and constraints (56) guarantee the right order of loading with respect to the bottom and the top slot of each platform. Constraints (57) synchronize the b_{qt} with the z_{it} variables.

4.7. Different variants of the formulations

The above presented four general formulations all consider two-way distance cost. Based on results reported in the literature (Boysen et al., 2010), one can expect those formulations to be intractable for realistic size instances. We therefore consider two simplified objective functions.

The first one considers solely the cost when the handling equipment is loaded (one-way distance). This simplifying assumption is motivated by the fact that one-way cost strongly correlates with the cost of the overall loading effort (Boysen et al., 2010). In this case, we can fix the value of the variables u_{qi} to 1 and remove these variables from the objective function. Accordingly, the constraints (23), (33), (41), and (53) are obsolete.

The second objective function makes an even stronger simplifying assumption. Namely, the distance cost is completely ignored in the objective function. In this case, we remove the term from the objective function and obtain:

$$\min \sum_{i \in N} \pi_i \left(1 - \sum_{q \in Q} v_{iq}\right) + \sum_{j \in J} \tau_j \sum_{k \in K_j} w_{jk} + \sum_{i \in N} \left(\beta^2 d_i + \beta^3 \gamma_i\right). \tag{58}$$

This may be an appropriate objective function for small and dense intermodal terminals where the cost associated with distance is negligible compared to the rehandling cost. We note that this formulation still allows to penalize detours.

4.8. Solution procedure

To accelerate the solution process, we use a heuristic to find an initial feasible loading sequence based on an optimal load plan and use it to warm start the general-purpose solver. The solution procedure is as follows:

1. Solve the LPP described by Mantovani et al. (2018) to optimality using a general-purpose solver.

2. Append the term for the one-way distance cost to the objective function and add a cut ensuring that the commercial value found in Step 1 is met.

3. Using a general-purpose solver, solve to optimality the problem of Step 2 providing the solution found in Step 1.

4. Calling Algorithm 1, determine a feasible loading sequence for the solution found in Step 3.

5. Add the sequencing constraints to the model, the additional costs to the objective function, and a cut for the minimum distance found in Step 3.

6. Solve the model of Step 5 providing the solution found in Step 4 as a warm start.

```
Data: Load plan
Result: Loading sequence
while not all assigned containers loaded do
    if from front side accessible container found whose assigned slot is ready then
        load container ;
    else if from back side accessible container found whose assigned slot is ready then
        load container by making a detour ;
    else
        load random container from front side whose assigned slot is ready and rehandle all
          blocking containers ;
    update state of slots ;
```

Algorithm 1: Algorithm for obtaining a feasible loading sequence based on a load plan

5. Numerical experiments

The aim of the numerical experiments is to analyze the impact of the different formulations on the performance of a general-purpose solver. We report the results of an extensive numerical study demonstrating the strengths and weaknesses of the formulations. We separate the results according to the definition of the objective function as the formulations with two-way distance are only tractable for small instances. First, we report results for experiments with the formulations considering full distance cost. Second, in Section 5.3, we report results with formulations considering the one-way distance cost for the loaded moves. Third, in Section 5.4, we examine the impact of the distance term in the objective function by ignoring the related costs. In Section 5.5, we analyze the performance of the best performing formulations without distance cost on larger instances. Finally, in Section 5.6, we analyze the benefit of solving the integrated LPSP compared to a sequential solution approach. For all experiments, we compare the results for two types of handling equipment: a GC and a RS. The experimental setup and test instances are described in the following section.

5.1. Test instances and experimental setup

The results are based on five sets of 50 instances each that are generated based on real data from our industrial partner.

Terminal layout and handling equipment. We consider a fixed storage area layout depicted in Figure 1: The containers are stacked at a maximum height of three containers and there is a maximum of six lots in total. Depending on the number of containers in a given instance, the number of stacks in each lot varies (it can take the values 1, 3, 5, 7, and 9). This is the layout of one of the industrial partner's terminals and it has been designed for RS vehicles. The results are computed for one specific make of RS and the actual restrictions related to reach and weight lifting capacity. For the sake of illustration, we also compute results for a GC. Note, however, that the layout remains the same and has hence not been designed for that type of handling equipment.

Instances. We generate 50 instances for different block sizes, ranging from what we refer to as extra small (XS) comprising only 15 containers to extra large (XL) comprising 150 containers. Table 3 provides an overview. Each instance is defined by the set of containers (each container is characterized by its size, height, weight and position in the yard) and the set of railcars (each railcar is characterized by its type and position on the track). For each of the set of instances of a given size, the generation is done in the same way. We start by describing the generation of sets of containers.

In the North American domestic market 53 ft containers are used while the international market is limited to mostly 20 and 40 ft containers. We therefore create two types of container sets: one with only 40 ft containers and one with a mix of 40, 45 and 53 ft containers. We note that we exclude 20 ft containers as they need to be stacked separately and also must be loaded in bottom slots. Hence, they are not challenging to deal with from a loading perspective. Moreover, we exclude 48 ft containers as they are treated like 53 ft ones. Similar to Mantovani et al. (2018), we draw instances at random from empirical distributions: We draw five instances of 40 ft containers and five instances with a mix of sizes. The latter leads to an average mix of 78% 40 ft, 8% 45 ft and 14% 53 ft. Container weights are drawn from an empirical distribution conditional on container size, and the position in the terminal is drawn at random. This results in 10 sets of containers for each instance size (50 in total).

The generated sets of containers are combined with five railcar sets, which leads to a total of 50 instances of each size. The railcars are drawn from the real North American fleet of railcars according to the same stratified random sampling protocol as in Mantovani et al. (2018). More precisely, the sequence of railcars is generated by adding a randomly drawn railcar until the given block length is reached.

Set name	# containers	Block length [ft]	# instances 40 ft cont.	# instances 40, 45, 53 ft cont.	# instances total
XS	15	200	25	25	50
S	50	667	25	25	50
M	75	1,000	25	25	50
L	113	1,500	25	25	50
XL	150	2,000	25	25	50

Table 3: Overview of the instances

Cost parameters. Making sure that the commercial value of a container always exceeds its loading cost, we set the cost parameters as follows: The commercial value (π_i) is 200 for all containers $i \in N$ as we do not focus on assessing the impact of container priorities. Costs for rehandlings (β^2) and detours (β^3) are set to 80 and 60, respectively. The setup costs (τ_j) are 1 for $j \in J$ and the distance cost between the stacking position of container i and slot q is $\beta^1_{iq} = |X^q - X^i| + Y^i$. The values of the parameters have been selected through a solution validation process with the industrial partner. These values have been selected so that they produce solutions with the desired trade-off between, e.g., rehandling, distance and detours. The values do not correspond to actual costs.

Implementation and solver. For the experiments, we use the general-purpose solver IBM ILOG CPLEX 12.8 on one thread of a 3.07 GHz processor equipped with 96 GB of RAM. The computational time limit is set to 36,000 seconds. This time limit exceeds the time budget for our problem which is maximum 30 minutes. However, we use this value for the sake of a better comparison of the models. The C++ language has been used for data handling, building the model, calling CPLEX, and running the algorithm. The reported computational times refer only to Step 6 of Algorithm 1 as the other times do not exceed a few seconds.

5.2. Experiments with full distance cost

In this section, we report results for the most challenging formulations with full consideration of distance cost (Sections 4.3 - 4.6) for the smallest instances (XS). The numerical results for the optimization are reported in Table 4. The computational time (CPU) indicates the average solution time for the instances that are solved to optimality. The average optimality gap is reported for instances that cannot be solved to optimality in the given time limit. For all formulations, no optimal solution can be found for any of the GC instances. For the RS instances, a small share can be solved to optimality. The average optimality gap is large (44.2%) even after ten hours of computation time. In 395 cases out of the total of 400 runs, a solution is found without any unproductive movement (rehandlings and detours). The remaining terms of the objective function are responsible for the large gaps.

Formulation	A1	A2	B1	B2	A1	A2	B1	B2
Handling equipment	GC	GC	GC	GC	RS	RS	RS	RS
# opt. solved instances	0	0	0	0	1	1	7	5
Avg. CPU [s]	-	-	-	-	333	590	553	683
Avg. opt. gap [%]	43.6	44.1	44.0	43.7	44.5	46.9	42.6	43.8

Table 4: Computational results for 50 XS instances (full distance)

These results show that even for small instances, the problem taking into account the full distance cost is intractable for a general-purpose solver. In the next section, we simplify the objective function and consider one-distance cost as proposed in Section 4.7. Boysen et al. (2010) show the cost for the loaded one-way moves strongly correlates with the cost of the overall effort, thus making this simplification reasonable. In our case, we observe a difference in transportation costs varying between 0.0% and 21.4% with an average of 6.6% when comparing the solutions of the XS instances solved to optimality with the two-way formulations to those obtained with the one-way formulation completed by the costs for unloaded moves in a post-processing step. While this number is based on few data points, it is in accordance with the findings of Boysen et al. (2010).

5.3. Experiments with one-way distance cost

In this section, we report results for the formulations with simplified one-way distance cost as described in Section 4.7 for instances of size XS and S. We start by analyzing the results for the smallest instances. Table 5 displays the average number of constraints and variables for all formulations, both one-way and two-way. We note that the number of constraints is drastically reduced by simplifying the distance cost.

Size	Distance cost	Handling equipment	A1	A2	B1	B2	A1	A2	B1	B2
			Variables				Constraints			
XS	one-way	GC & RS	5,787	5,744	3,680	3,680	1,275	3,487	3,409	3,610
XS	two-way	GC & RS	5,906	5,906	3,842	3,842	34,288	36,499	34,153	34,354
S	one-way	GC	86,148	86,148	14,367	14,367	6,077	79,384	78,638	80,918
S	one-way	RS	86,215	86,215	14,434	14,434	20,817	94,123	93,378	95,658

Table 5: Average number of variables and constraints per formulation depending on size, distance cost, and handling equipment

The results for XS instances are reported in Table 6 showing that all instances can be solved to optimality within short computational time (average 11 and 29 seconds for GC and RS movements, resp.). Three out of 50 RS instances require one rehandled container, one of them additionally a detour. All other instances are solved without any unproductive movement.

Formulation	A1	A2	B1	B2	A1	A2	B1	B2
Handling equipment	GC	GC	GC	GC	RS	RS	RS	RS
# opt. solved instances	50	50	50	50	50	50	50	50
Avg. CPU [s]	3	31	6	4	18	86	6	7

Table 6: Computational results for the 50 XS instances (one-way distance)

We now turn our attention to S instances. To give an idea about the sizes of the different formulations, we report the average number of variables and constraints in Table 5. The number of variables is much higher for the A formulations (average 86,182) than for the B formulations (14,400). The difference in the number of variables related to the handling equipment is small. The number of constraints is the lowest for the formulation A1 (6,077 for GC, and 20,817 for RS) and ranges between 78,638 and 95,658 for the other formulations. Comparing the RS to the GC movements, the number of constraints increases by roughly 15,000 on average.

We report the results for S instances in Table 7. In addition to computing times and optimality gap, we report the average number of rehandlings and detours for instances that could not be solved to optimality. The results show that the problems with RS movements are harder to solve than those with GC movements: the average number of optimally solved instances drops from 41 to 32, the average computational time rises from roughly 70 minutes to 160 minutes and the average optimality gap increases from 8% to 18%. This is related to the higher number of forbidden loading sequences. For both types of handling equipment, all optimal solutions are without unproductive movements.

For the GC, the number of instances solved to optimality varies between 37 and 43 out of a total of 50. The formulation B1 finds the highest number of optimal solutions, followed

Formulation Handling equipment	A1 GC	A2 GC	B1 GC	B2 GC	A1 RS	A2 RS	B1 RS	B2 RS
# opt. solved instances	42	37	43	42	26	22	41	40
Avg. CPU [s]	5,418	9,268	1,423	821	16,897	19,462	1,388	1,041
Avg. opt. gap [%]	25.5	5.0	0.6	0.6	61.4	7.7	0.6	0.6
Avg. # rehandlings (timeout)	1.5	0.2	0.0	0.0	4.5	0.1	0.0	0.0
Avg. # detours (timeout)	-	-	-	-	2.4	0.2	0.0	0.0

Table 7: Computational results for the 50 S instances (one-way distance)

by formulations A1 and B2. Concerning the computational times, the B formulations clearly outperform the others. Regarding the optimality gaps, the formulation A1 is worst. The average number of rehandled containers for instances that are not optimally solved at timeout is rather low for all formulations.

The RS instances, however, show a different picture. Whereas the number of optimally solved instances decreases significantly for the A formulations, the numbers remain almost stable for the B formulations. The latter clearly surpass the other formulations with respect to computational time and optimality gap. Comparing the average number of unproductive movements at timeout, they are reasonable for all formulations and lowest for A2, B1, and B2.

We now examine whether and how the container characteristics have an impact on the computational results. Figure 4 shows a comparison of the number of optimally solved instances for the two settings of container characteristics (40 ft only and mixed 40, 45 and 53 ft). With only one exception (A1 formulation and RS), the share of solved instances with a mix of container sizes is higher or equal (average of 24 for GC, 18 for RS). Due to the symmetry of load plans, the solution space for containers of equal size is larger. We also note the difference between the two types of handling equipment when dealing with containers of equal size: On average 18 out of 25 instances can be solved to optimality in the case of GCs. The average number for a RS is 15.

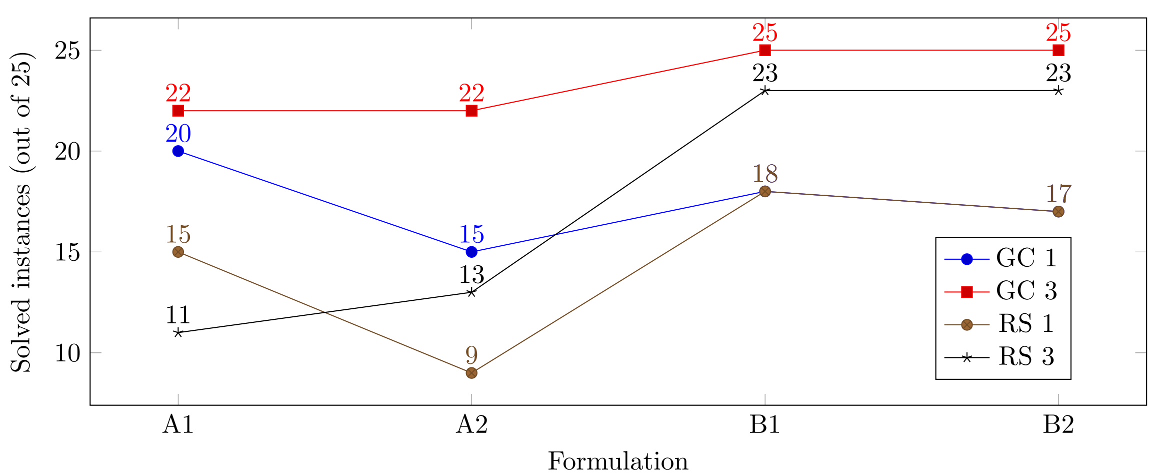

Figure 4: Number of solved S instances with respect to container characteristics for each formulation (1 size: 40 ft, 3 sizes: 40 ft, 45 ft and 53 ft)

In summary, the B formulations generally exhibit the best performance on S instances in terms of average computational time (below the 30 minute time budget) and number of optimally solved instances. The reduction in the number of variables at the expense of more constraints seems to be the best trade-off we could find. Interestingly, these formulations seem to be hardly affected by the higher complexity caused by the RS movements. Even though the B formulations work best on average in terms of computational time and number of optimally solved instances, there is no clear dominance relationship between the formulations. Depending on the actual settings of the terminal where the model can be applied, e.g., gantry crane and homogeneous containers, one or the other formulation may be the best choice. The two different definitions of the z variables (i loaded *at* or *by* t) seem not to affect the performance strongly, but with respect to the solution quality at timeout, A2 dominates A1.

5.4. Experiments ignoring distance cost

In this section, we present experiments ignoring cost for distance (as described in Section 4.7) and the results are reported in Table 8. Similarly to the results presented in Section 5.3, the RS instances are harder to solve. While all GC instances could be solved to optimality, there are 2 (resp. 1) instances that cannot be solved to optimality by the formulation A1 (resp. A2). Compared to the results with one-way distance cost (41 and 32 optimally solved instances), these figures are considerably higher.

Formulation Handling equipment	A1 GC	A2 GC	B1 GC	B2 GC	A1 RS	A2 RS	B1 RS	B2 RS
# opt. solved instances	50	50	50	50	48	47	50	50
Avg. CPU [s]	1,091	3,969	651	995	8,935	11,546	1,210	1,361
Avg. opt. gap [%]	-	-	-	-	100.0	95.5	-	-
Avg. # rehandlings (timeout)	-	-	-	-	16.5	1	-	-
Avg. # detours (timeout)	-	-	-	-	10	1	-	-

Table 8: Computational results for the 50 S instances (no distance)

For the GC, the average computation time drops from roughly 90 (resp. 155) to 18 (resp. 66) minutes for the formulation A1 (resp. A2) compared to the prior experiments (Section 5.3). The computational times for the formulations B1 and B2 are on average 11 and 17 minutes, respectively.

The RS movements intensify the differences between the formulations. The B formulations are the only ones for which one can solve all 50 instances to optimality. The average computation time is 20 minutes for B1 and 23 minutes for B2, respectively. The formulations A1 and A2 solve 48 and 47 instances, respectively. The computational times are 149 and 192 minutes, and hence much higher compared to B1 and B2.

The optimality gap for instances without proven optimum is large ($> 95\%$). The number of unproductive movements at timeout is reasonable for the instances solved by formulation A2. In terms of computational time, the B formulations clearly outperform the others, but the A formulations can still deal with the instance size.

The figures show that the cost for the distance in the objective function makes the optimization problem considerably harder. Therefore, in a terminal where distance is not a major concern, it can be valuable to discard this term from the objective function. We note that it does not prevent the use of other proxies for distance, such as the one we use for detours.

5.5. Experiments with larger instances

The computational experiments reported in Sections 5.3 and 5.4 indicate that the formulations B1 and B2 work best with respect to the number of optimally instances solved in the given time limit and computational time. We now use these formulations without consideration of the distance on larger instances (sizes M, L, and XL) to examine which instance size can be solved in reasonable time. The results are reported in Table 9. Recall that the largest instances that are solved to optimality for the LPSP for single-stack trains in a similar setting comprise 40 containers (Ambrosino et al., 2013).

Instance size (# containers)	S (50)		M (75)		L (113)		XL (150)	
Formulation	B1	B2	B1	B2	B1	B2	B1	B2
				GC				
# opt. solved instances	50	50	50	50	11	15	1	0
Avg. CPU [s]	651	995	3,333	2,688	31,000	22,777	15	-
Avg. opt. gap [%]	-	-	-	-	99.6	95.8	99.9	99.8
Avg. # rehandlings (timeout)	-	-	-	-	25.5	16.3	39.6	38.9
				RS				
# opt. solved instances	50	50	28	36	0	0	5	0
Avg. CPU [s]	1,210	1,361	19,727	17,450	-	-	20	-
Avg. opt. gap [%]	-	-	98.2	95.9	100.0	99.8	100.0	100.0
Avg. # rehandlings (timeout)	-	-	14.3	3.7	65.6	60.6	88.0	82.8
Avg. # detours (timeout)	-	-	2.0	2.9	7.0	7.2	7.5	6.7

Table 9: Computational results for the large instances (50 of each size) without consideration of distance cost

The share of instances solved to optimality within ten hours drops significantly with increasing size. For all sizes, no unproductive movements are part of the solutions of those instances that are solved to optimality. All 50 GC instances of size M (75 containers) can be solved to optimality. For the RS instances, however, only 28 of the instances can be solved to optimality for the B1 and 36 for the B2 formulations. The computational times are above five hours and thus not appropriate for operational problems.

Larger instances (size L) comprising a block of at least 1,500 ft length and 113 containers are hard. Only 11 (resp. 15) instances are solved optimally with the formulation B1 (resp. B2) within the given time limit. None of the RS instances can be solved to optimality. Note that the average optimality gap by the end of the computational time is large ($> 95\%$). The solution quality in terms of unproductive movements at timeout is poor except for the B2 formulation with instance size M.

The largest instances that we test (size XL) comprise 2,000 ft and 150 containers. Only few instances can be solved to optimality. In those cases, the surprisingly fast computational times are related to the fact that the heuristic provides an initial optimal solution (without any unproductive movements) and CPLEX quickly proves the optimality of the solution. In none of the cases could the solver find the optimal solution without the initialization.

5.6. Savings achieved by solving the integrated LPSP

Last, we report results from a study investigating whether the integrated LPSP can reduce the handling cost compared to the sequential solution as suspected in our motivation. We therefore solve twice the same S instances with consideration of the one-way distance cost. In the first run, we optimize the load plan. In the second run, we optimize the load sequencing taking the fixed load plan as an input. Accordingly, the result of the first run corresponds to the outcome of the methodology presented by Mantovani et al. (2018) with an additional summand in the objective function related to the distance cost.

As not all instances can be solved to optimality with the integrated model (Table 7), we only compare instances that are optimally solved with at least one of the four formulations (49 GC instances, 41 RS instances). The distribution of the number of rehandled containers obtained by the sequential model is displayed in Figure 5.

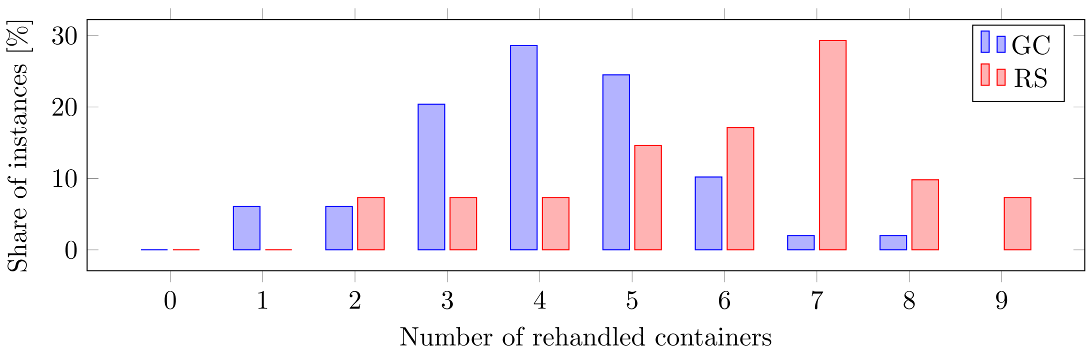

Figure 5: Distribution of the number of rehandled containers for the sequential solving of the LPSP

For the GC experiments, up to 8 containers (average 4.1) must be rehandled in the sequential model, whereas the number of rehandlings can be reduced to 0 with the integrated model. For the RS instances, the number of rehandled containers varies between 2 and 9 (average 5.9) and drops to 0. The number of detours varying between 0 and 2 (average 0.4) can be lessened to 0. In other words, the integrated model could find for every single instance a loading sequence without using any unproductive movement at all. On average, the load plans comprise 32 containers corresponding to 32 handlings for the integrated model. Compared to 36.1 and 38.3 handlings, a saving of 11.3% can be achieved for the GC and 16.5% for the RS instances, respectively. The average improvement on the objective function is 7.6% and 8.1%, respectively. Recall that the solution of the integrated model is in no case worse than the solution found in the sequential procedure in terms of penalty for not loaded containers.

The integrated model tends to choose load plans that either include all containers of one stack or those that are most easily accessible. Contrarily to the load plans found without consideration of the handling cost, containers whose neighboring load units in the pile are not loaded are rarely part of the load plan.

These results underline that, from an operational point of view, the solution quality can be significantly improved with an integrated model. Hence, terminal operators could clearly benefit from a lower handling cost. These benefits become even clearer if we consider that the loading is executed for many blocks a day and that the handling equipment is one of the most critical resources in terminals.

6. Conclusions and future research

In this paper, we have introduced the load planning and sequencing problem for double-stack intermodal trains. We have modeled the movements of the handling equipment in a preprocessing step, such that a set of forbidden retrieval sequences is obtained. Starting from the model for the load planning problem for double-stack intermodal trains proposed by Mantovani et al. (2018), we have introduced four different ILP formulations.

Computational results related to one block of an intermodal train showed that the instances considering reach stackers instead of gantry cranes are more difficult to solve. This is due to more complex accessibility rules of the reach stackers yielding more dependencies between the movements, which finally results in a higher number of constraints. Furthermore, we found that, on average, the formulations B1 and B2 work best for our instances. Both of them introduce two sets of decision variables: one is related to the perspective of the containers z_{it} and the second one to the perspective of slots of the train b_{qt}. These models are less affected by the higher number of forbidden sequences caused by the reach stacker. However, this does not reflect a clear dominance relationship between the formulations. Depending on the instance characteristics, the formulation A1 might work better than the B formulations. Applied in practice, a parallel optimization of multiple models might be appropriate to reliably and quickly find solutions for the LPSP.

Additionally, we showed that by ignoring the costs for the distances occurring in the terminal, as it may be suitable for terminals with a compact layout, we could solve instances with a block length of 1,000 ft and 75 containers with a commercial solver to optimality. This is a significant improvement compared to the literature, where the largest optimally solved instances in a similar setting for single-stack railcars comprise 40 containers (Ambrosino et al., 2013). For larger instances, however, the computational times are too high for an operational problem. Comparing the computational results to those reported by Mantovani et al. (2018), the remarkable increase in complexity by integrating the load planning and the load sequencing problem becomes clear.

Finally, we highlighted that the integrated model can significantly reduce the handling cost in terminals compared to the sequential solving. In our case study comprising 50 containers, the number of rehandled containers drops from 4.1 to 0 for gantry crane movements and from 5.9 to 0 for reach stacker movements. As the problem is solved many times per day in a terminal, this can lead to a significant decrease in handling costs.

Future research should be dedicated to an alternative approach to model the two-way distance to achieve a more tractable formulation. In addition, the sequencing of rehandles is a relevant extension of the model as, in rare cases, the rehandling of blocking containers may involve further rehandlings. Tailored solution methods and valid inequalities for the models are an additional subject for future research as fast solutions need to be found by terminal operators. Last, it might be interesting to take more than one handling equipment into account at once.

Acknowledgments

We gratefully acknowledge the close collaboration with the Canadian National Railway Company (CN) and the funding through the CN Chair in Optimization of Railway Operations. We also acknowledge the support of the German Academic Exchange Service (DAAD) for the first author's research stay at CIRRELT (Montreal, Canada). We are grateful to four anonymous referees for their valuable comments which helped to improve the quality of this paper.

References

Adobe Stock (2020a). Container vagoons in export and import business and logistics. Aerial view. URL: https://stock.adobe.com/ca/images/container-vagoons-in-export-and-import-business-and-logistics-aerial-view/288509635.

Adobe Stock (2020b). Kontenery. URL: https://stock.adobe.com/ca/images/kontenery/121855329.

Ahuja, R. K., Jha, K. C., & Liu, J. (2007). Solving Real-Life Railroad Blocking Problems. *Interfaces*, *37*, 404–419. doi:10.1287/inte.1070.0295.

Ambrosino, D., Bernocchi, L., & Siri, S. (2016). Multi-Objective Optimization for the Train Load Planning Problem with Two Cranes in Seaport Terminals. *IFAC-PapersOnLine*, *49*, 383–388. doi:10.1016/j.ifacol.2016.07.064.

Ambrosino, D., Bramardi, A., Pucciano, M., Sacone, S., & Siri, S. (2011). Modeling and Solving the Train Load Planning Problem in Seaport Container Terminals. In *IEEE International Conference on Automation Science and Engineering* (pp. 208–213). Trieste. doi:10.1109/CASE.2011.6042439.

Ambrosino, D., & Caballini, C. (2018). New Solution Approaches for the Train Load Planning Problem. *EURO Journal on Transportation and Logistics*, (pp. 1–27). doi:10.1007/s13676-018-0127-x.

Ambrosino, D., Caballini, C., & Siri, S. (2013). A Mathematical Model to Evaluate Different Train Loading and Stacking Policies in a Container Terminal. *Maritime Economics and Logistics*, *15*, 292–308. doi:10.1057/mel.2013.7.

Ambrosino, D., & Siri, S. (2014). Models for Train Load Planning Problems in a Container Terminal. In R. de Sousa, Jorge Freire and Rossi (Ed.), *Computer-Based Modelling and Optimization in Transportation* chapter Advances in Intelligent Systems and Computing. (pp. 15–25). Springer volume 262. doi:10.1007/978-3-319-04630-3_2.

Ambrosino, D., & Siri, S. (2015). Comparison of Solution Approaches for the Train Load Planning Problem in Seaport Terminals. *Transportation Research Part E: Logistics and Transportation Review*, *79*, 65–82. doi:10.1016/j.tre.2015.04.003.

Anghinolfi, D., Caballini, C., & Sacone, S. (2014). Optimizing Train Loading Operations in Innovative and Automated Container Terminals. In *IFAC Proceedings Volumes* (pp. 9581–9586). Cape Town, South Africe volume 19. doi:10.3182/20140824-6-ZA-1003.02553.

Association of American Railroads (2017). Loading Capabilities Guide. URL: https://www.aar.com/standards/pdfs/AAR%20Loading%20Capabilities%20Guide%202017.pdf.

Bierwirth, C., & Meisel, F. (2010). A Survey of Berth Allocation and Quay Crane Scheduling Problems in Container Terminals. *European Journal of Operational Research*, *202*, 615–627. doi:10.1016/j.ejor.2009.05.031.

Bostel, N., & Dejax, P. (1998). Models and Algorithms for Container Allocation Problems on Trains in a Rapid Transshipment Shunting Yard. *Transportation Science*, *32*, 370–379. doi:10.1287/trsc.32.4.370.

Boysen, N., Briskorn, D., & Meisel, F. (2017). A Generalized Classification Scheme for Crane Scheduling with Interference. *European Journal of Operational Research*, *258*, 343–357. doi:10.1016/j.ejor.2016.08.041.

Boysen, N., & Fliedner, M. (2010). Determining Crane Areas in Intermodal Transshipment Yards: The Yard Partition Problem. *European Journal of Operational Research*, *204*, 336–342. doi:10.1016/j.ejor.2009.10.031.

Boysen, N., Fliedner, M., Jaehn, F., & Pesch, E. (2013). A Survey on Container Processing in Railway Yards. *Transportation Science*, *47*, 312–329. doi:10.1287/trsc.1120.0415.

Boysen, N., Fliedner, M., & Kellner, M. (2010). Determining Fixed Crane Areas in Rail-Rail Transshipment Yards. *Transportation Research Part E: Logistics and Transportation Review*, *46*, 1005–1016. doi:10.1016/j.tre.2010.05.004.

Bruns, F., Goerigk, M., Knust, S., & Schöbel, A. (2014). Robust Load Planning of Trains in Intermodal Transportation. *OR Spectrum*, *36*, 631–668. doi:10.1007/s00291-013-0341-8.

Bruns, F., & Knust, S. (2012). Optimized Load Planning of Trains in Intermodal Transportation. *OR Spectrum*, *34*, 511–533. doi:10.1007/s00291-010-0232-1.

Carlo, H. J., Vis, I. F. A., & Roodbergen, K. J. (2014a). Storage Yard Operations in Container Terminals: Literature Overview, Trends, and Research Directions. *European Journal of Operational Research*, *235*, 412–430. doi:10.1016/j.ejor.2013.10.054.

Carlo, H. J., Vis, I. F. A., & Roodbergen, K. J. (2014b). Transport Operations in Container Terminals: Literature Overview, Trends, Research Directions and Classification Scheme. *European Journal of Operational Research*, *236*, 1–13. doi:10.1016/j.ejor.2013.11.023.

Cichenski, M., Jaehn, F., Pawlak, G., Pesch, E., Singh, G., & Blazewicz, J. (2017). An Integrated Model for the Transshipment Yard Scheduling Problem. *Journal of Scheduling*, *20*, 57–65. doi:10.1007/s10951-016-0470-4.

Corry, P., & Kozan, E. (2006). An Assignment Model for Dynamic Load Planning of Intermodal Trains. *Computers & Operations Research*, *33*, 1–17. doi:10.1016/j.cor.2004.05.013.

Corry, P., & Kozan, E. (2008). Optimised Loading Patterns for Intermodal Trains. *OR Spectrum*, *30*, 721–750. doi:10.1007/s00291-007-0112-5.

Dotoli, M., Epicoco, N., Falagario, M., Palma, D., & Turchiano, B. (2013). A Train Load Planning Optimization Model for Intermodal Freight Transport Terminals: A Case Study. In *Proceedings - 2013 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2013* (pp. 3597–3602). doi:10.1109/SMC.2013.613.

Dotoli, M., Epicoco, N., Falagario, M., Seatzu, C., & Turchiano, B. (2017). A Decision Support System for Optimizing Operations at Intermodal Railroad Terminals. In *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (pp. 1–15). volume 47. doi:10.1109/TSMC.2015.2506540.

Dotoli, M., Epicoco, N., & Seatzu, C. (2015). An Improved Technique for Train Load Planning at Intermodal Rail-Road Terminals. In *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*. doi:10.1109/ETFA.2015.7301580.

Dyckhoff, H. (1990). A Typology of Cutting and Packing Problems. *European Journal of Operational Research*, *44*, 145–159. doi:10.1016/0377-2217(90)90350-K.

Expósito-Izquierdo, C., Melián-Batista, B., & Moreno-Vega, J. M. (2015). An Exact Approach for the Blocks Relocation Problem. *Expert Systems with Application*, *42*, 6408–6422. doi:10.1016/j.eswa.2015.04.021.

Heggen, H., Braekers, K., & Caris, A. (2016). Optimizing Train Load Planning: Review and Decision Support for Train Planners. In A. Paias, M. Ruthmair, & S. Voß (Eds.), *Computational Logistics* (pp. 193–208). 7th International Conference, ICCL 2016 Springer volume 9855 LNCS. doi:10.1007/978-3-319-44896-1_13.

Lai, Y.-C., Barkan, C. P. L., & Onal, H. (2008a). Optimizing the Aerodynamic Efficiency of Intermodal Trains. *Transportation Research Part E*, *44*, 820–834. doi:10.1016/j.tre.2007.05.011.

Lai, Y.-C., Ouyang, Y., & Barkan, C. P. L. (2008b). A Rolling Horizon Model to Optimize Aerodynamic Efficiency of Intermodal Freight Trains with Uncertainty. *Transportation Science*, *42*, 466–477. doi:10.1287/trsc.1080.0245.

Lang, M. X., Zhou, X., & Sun, Y. (2015). Multi-Objective Optimization for Double Stack Train Loading Problem. *Jiaotong Yunshu Xitong Gongcheng Yu Xinxi/Journal of Transportation Systems Engineering and Information Technology*, *15*, 94–100 and 106.

Mantovani, S., Morganti, G., Umang, N., Crainic, T. G., Frejinger, E., & Larsen, E. (2018). The Load

Planning Problem for Double-Stack Intermodal Trains. *European Journal of Operational Research*, *267*, 107–119. doi:10.1016/j.ejor.2017.11.016.

Morganti, G., Crainic, T. G., Frejinger, E., & Ricciardi, N. (2019). *Block Planning for Intermodal Rail: Methodology and Case Study*. Technical Report CIRRELT.

Otto, A., Li, X., & Pesch, E. (2017). Two-Way Bounded Dynamic Programming Approach for Operations Planning in Transshipment Yards. *Transportation Science*, *51*, 325–342. doi:10.1287/trsc.2016.0688.

Pacanovsky, D. L., Jahren, C. T., Palmer, R. N., Newman, R. R., & Howland, D. (1995). A Decision Support System to Load Containers to Double-Stack Rail Cars. *Civil Engineering Systems*, *11*, 247–267. doi:10.1080/02630259508970149.

Souffriau, W., Vansteenwegen, P., Vanden Berghe, G., & Van Oudheusden, D. (2009). Variable Neighbourhood Descent for Planning Crane Operations in a Train Terminal. In K. Sörensen, M. Sevaux, W. Habenicht, & M. J. Geiger (Eds.), *Metaheuristics in the Service Industry* (pp. 83–98). Berlin Heidelberg: Springer volume 624 of *Lecture Notes in Economics and Mathematical Systems*. doi:10.1007/978-3-642-00939-6_6.

Stahlbock, R., & Voß, S. (2008). Operations Research at Container Terminals: A Literature Update. *OR Spectrum*, *30*, 1–52. doi: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 Spectrum*, *26*, 3–49. doi:10.1007/s00291-003-0157-z.

Wang, L., & Zhu, X. (2014). Rail Mounted Gantry Crane Scheduling Optimization in Railway Container Terminal Based on Hybrid Handling Mode. *Computational Intelligence and Neuroscience*, *2014*, 1–8. doi:10.1155/2014/682486.

Wang, Z., Lang, M., Zhou, X., Przybyla, J., & Sun, Y. (2018). A Tabu Search Algorithm for Loading Containers on Double-Stack Cars. In F. Xhafa, S. Patnaik, & A. Y. Zomaya (Eds.), *Advances in Intelligent Systems and Interactive Applications. Proceedings of the 2nd International Conference on Intelligent and Interactive Systems and Applications (IISA2017)* (pp. 49–56). Springer. doi:10.1007/978-3-319-69096-4.

Wäscher, G., Haußner, H., & Schumann, H. (2005). An Improved Typology of Cutting and Packing Problems. *European Journal of Operational Research*, *183*, 1109–1130. doi:10.1016/j.ejor.2005.12.047.

Zhang, C., Liu, J., wah Wan, Y., Murty, K. G., & Linn, R. J. (2003). Storage Space Allocation in Container Terminals. *Transportation Research Part B*, *37*, 883–903. doi:10.1016/S0191-2615(02)00089-9.