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. Even though this operational problem is highly 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. We introduce a flexible modelling of the movements of the handling equipment adaptable to various operating conditions. By extending prior work on load planning, we propose six integer linear programming formulations differing in the number of constraints and variables. An extensive numerical study identifies two better performing formulations. With these formulations, we solve medium-size instances with a commercial general-purpose solver in reasonable time. A case study based on real data from the North American market highlights that the integrated load planning and sequencing problem can considerably reduce the container handling cost in intermodal terminals compared to sequential solutions.

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 intermodal freight, load units are transported from origin to destination using at least two different modes of transportation. The long-haul leg of ground transportation is typically carried out by rail. Therefore, load units – usually in the form of standardized containers – must be transferred from one transportation mode to another. This takes place in intermodal terminals, which are crucial for the efficiency of the overall transport chain.

In intermodal terminals, many difficult planning problems arise on the strategic, tactical, and operational levels. This paper addresses an operational problem which consists in determining the containers to be loaded on a double-stack train along with the sequence of retrieving and loading moves such that the value of the loaded containers is maximized and

*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)

the handling effort in the terminal is minimized. We refer to this problem as the load planning and sequencing problem (LPSP) for double-stack intermodal trains.

Even though this problem is highly integrated, it has mostly been treated as two separate problems in the related literature. The load planning problem finds a subset of stored containers to be loaded on a train and the exact way of loading them to minimize the value of unloaded containers while obeying all relevant technical constraints. The so-called load plan then imposes restrictions on the load sequencing problem, whose aim is to determine a loading order of containers meeting all relevant constraints and minimizing the handling cost in the terminal.

Solving these problems sequentially might lead to unfavourable results: the load plan is determined without taking sequencing constraints into account and the load sequencing is based on a fixed load plan. Considering two containers allocated to the same railcar one on top of the other, it is necessary to place the container assigned to the bottom slot onto the railcar first. If this particular container cannot be retrieved from the stack before the second container, unproductive handling movements inevitably take place. However, there are typically many different optimal load plans that comprise the same or a similar set of containers. Taking the sequencing constraints into account while determining the load plan, one could switch the container assignments and achieve a load plan with exactly the same commercial value, thus allowing a more efficient loading sequence. Integrating the load planning and the load sequencing problems can therefore reduce the loading effort without deteriorating the load plan.

The integrated problem is, however, very challenging due to many complex loading rules depending on railcar and container characteristics connected with varying retrieving sequences determined by crane types, container positions and characteristics. By contrast to single-stack railcars, the sequencing does not only depend on the stacking position but also on the current loading state of each railcar. This is why even small changes in the load plan can have a large impact on the load sequencing. In addition, we consider an operational problem which should be solved within minutes.

Most of the existing literature related to the load planning problem considers single-stack railcars (e.g., Bruns and Knust, 2012; Dotoli et al., 2013), even though double-stacking of containers is of high relevance for the North American market. Mantovani et al. (2018) propose a novel model for the load planning problem 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 part of the problem. A few papers have addressed the integrated LPSP, namely Corry and Kozan (2006) and Ambrosino et al. (2011), among others. Even though they solely consider single-stack railcars, large instances cannot be solved by exact methods. Part of the models make simplifying assumptions such as pre-set loading sequences, the exclusion of rehandling movements, or the prohibition of overbooking. The largest instances solved to optimality without those simplifying assumptions comprise 40 containers (Ambrosino et al., 2013).

The contributions of this paper are threefold: First, we introduce six integer linear program (ILP) formulations for the LPSP for double-stack intermodal trains including a flexible way to model the local restrictions related to handling equipment and terminal layout. This problem is of high practical relevance, especially for rail freight on 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. Second, we numerically assess the performance of the

ILP formulations by conducting extensive numerical experiments both with gantry crane and reach stacker movements by applying an exact algorithm. We also evaluate different ways to integrate distance in the objective function. The experiments allow us to identify two better performing formulations, which introduce two sets of decision variables related to the loading sequence of the containers and to the loading state of the platforms. With these formulations, an optimal load and sequence plan can be reliably found for blocks of 1,000 ft and 75 containers. Third, we report results of a case study highlighting the benefits of our model compared to sequential models in terms of a significant reduction of the handling cost in terminals. In the test instances comprising 50 containers, the sequential solutions require on average 4.1 rehandled containers for gantry cranes and 5.9 for reach stackers, respectively. The rehandling of containers and other unproductive movements can be totally avoided for all instances when solving the integrated model.

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 LPSP for double-stack intermodal trains. 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 problems related to the planning and operation of container terminals (Vis and de Koster, 2003; 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 problem addressed in this paper comprises two out of five subproblems: deciding on the containers' positions on trains and on the sequence of container moves per crane. In spite of a multitude of papers on related problems, there is to the best of our knowledge no model considering the integrated LPSP for double-stack intermodal trains.

2.1. Load planning problem

As stated by Mantovani et al. (2018), the load planning problem for intermodal trains can be seen as a special case of the packing-cutting-knapsack problem (Dowsland and Dowsland, 1992; Dyckhoff et al., 1997; Martello and Toth, 1990; Wäscher et al., 2005). According to a common typology (Wäscher et al., 2005; Dyckhoff, 1990), the load planning problem 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 load planning problem has been extensively studied with different levels of detail. Some papers focus on particularly detailed modelling of 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). Double-stack trains are considered in a few models (Corry and Kozan, 2008; Bruns and Knust, 2012; Heggen et al., 2016; Lai et al., 2008a,b; Mantovani et al., 2018), however, 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). 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 setup costs (preparation of the railcar for a given combination of container types). 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. 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). Imai et al. (2006) investigate the simultaneous stowage and load planning for a container ship aiming at maximizing stability and minimizing the rehandling of containers in the yard. They propose a multi-objective integer programming formulation. 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 organizes the blocks such that the number of relocation movements found by the BRP is minimized (Expósito-Izquierdo et al., 2015). However, the loading sequence is an input and not subject to optimization for both problems. The load sequencing problem itself can be seen as a NP-hard asymmetric traveling salesman problem (Boysen et al., 2010).

Some settings conduct the load sequencing in a second stage 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). We, however, consider the integrated LPSP for double-stack intermodal trains. Similar integrated problems are addressed in some papers, but none of them permits double-stacking of containers on trains. 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 cranes.

Like all other papers apart from Corry and Kozan (2006), we assume that the train has been unloaded in a prior stage and the scope of the problem is limited to the loading process. The vast majority of the papers – including ours – assume that the sequencing problem is decomposable by crane and therefore consider one crane 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 papers involve more than one crane in the sequencing problem (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 papers 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 a non-sequential loading policy could not be solved to optimality, no consequences on the number of rehandlings are reported. Others, similar to ours, impose a non-sequential loading of the train (Ambrosino and Siri, 2015; Ambrosino et al., 2016; Corry and Kozan, 2006). All load planning and sequencing papers 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 papers, 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 reshuffles (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).

We consider here a static environment with deterministic data: the availability of containers and railcars does not change over time. By contrast, a dynamic setting with uncertainty in the data is treated by Corry and Kozan (2006). They adapt the load plan in a rolling horizon environment by solving a deterministic model with updated data each time a trigger event occurs.

Typical objectives of the LPSP are the minimization of the handling cost consisting of rehandlings (Corry and Kozan, 2006; Ambrosino et al., 2011, 2013; Ambrosino and Siri, 2014; Ambrosino et al., 2016; 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 the gantry crane along the track, whereas Ambrosino and Siri (2014) only consider unproductive backward movements of the crane. The latter two papers exclude the costs for the distance of reach stackers that place the containers next to the assigned railcar. Most papers additionally incorporate objectives related to the load planning problem discussed in Section 2.1.

Due to the complexity of the problem, exact algorithms are applied only to small instances. Some papers develop tailored heuristic solution techniques (Ambrosino et al., 2011; Ambrosino and Siri, 2015; Ambrosino and Caballini, 2018). The largest instances solved to optimality are relatively small compared to the load planning problem. Ambrosino et al. (2013) solve instances comprising 40 containers allowing a non-sequential loading order and rehandling of containers. Ambrosino and Siri (2015) solve instances with 50 containers imposing a strictly sequential loading order. Ambrosino et al. (2016) consider two cranes and solve instances with 24 containers to optimality. Recall that the former models do not consider double-stacking of containers.

2.3. Summary

Table 1 extends the literature review by Heggen et al. (2016) and summarizes the characteristics of the LPSP discussed earlier. As this literature review shows, and 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) is the only paper that takes a high variety of loading patterns dealing with double-stack railcars into account. However, this model misses the sequencing part. All integrated load planning and sequencing models only 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 of trains (Corry and Kozan, 2006).

	Problem		**Characteristics**				**Objective**				**Nature of the problem and solution method**		
	Load planning	Load sequencing	Weight restrictions	Double-stacking	Non-sequential loading	Rehandling	Train utilization	Setup costs	Handling costs	Other	Static / Dynamic (s/d)	Deterministic / Stochastic (d/s)	Exact / Heuristic (e/h)
Aggoun et al. (2011)	•		•				•				d	d	h
Ambrosino et al. (2011)	•	•	•			•	•		•		s	d	e + h
Ambrosino et al. (2013)	•	•	•		[•]	[•]	•		•		s	d	e
Ambrosino and Siri (2014)	•	•	•		[•]	[•]	•		•		s	d	e
Ambrosino and Siri (2015)	•	•	•		[•]	•	•		•		s	d	h
Ambrosino et al. (2016)	•	•	•		•	•	•	•	•		s	d	e
Ambrosino and Caballini (2018)	•	•	•			•	•		•		s	d	h
Anghinolfi et al. (2014)	•		•				•				s	d	e + h
Bostel and Dejax (1998)	•								•		d	d	e + h
Bruns and Knust (2012)	•		•	(•)			•	•			s	d	e
Bruns et al. (2014)	•		•				•	•		•	s	s	e + h
Cichenski et al. (2017)	•						•		•	•	s	d	e
Corry and Kozan (2006)	•	•			•	•			•	•	d	s	e
Corry and Kozan (2008)	•			(•)		•		•	•	•	d	s	h
Dotoli et al. (2013)	•		•				•			•	s	d	e
Dotoli et al. (2015)	•		•				•			•	s	d	h
Dotoli et al. (2017)	•		•				•			•	s	d	e
Heggen et al. (2016)	•		•	(•)			•			•	s	d	e
Lai et al. (2008a)	•		•	•						•	s	d	e
Lai et al. (2008b)	•		•	•						•	d	s	e
Mantovani et al. (2018)	•		•	•			•	•			s	d	e

Table 1: Related literature on the load planning and the load sequencing problem for intermodal trains
(•): model adaptable but not tested, [•]: depending on model in paper

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

The LPSP is a highly integrated operational problem governed by the characteristics of the layout of the terminal, the containers, the train, and the handling equipment.

3.1. Intermodal rail-road terminals

An intermodal rail-road terminal consists of several areas. A schematic layout as considered in this paper is depicted in Figure 1. Trucks arrive at the terminal to unload their load units. After being picked up by the handling equipment, the load units are stored in the storage area. Direct rail-road or rail-rail transfers are not considered in this paper. Cranes and trucks may move in the gray zones (Figure 1), but terminal-specific restrictions may apply.

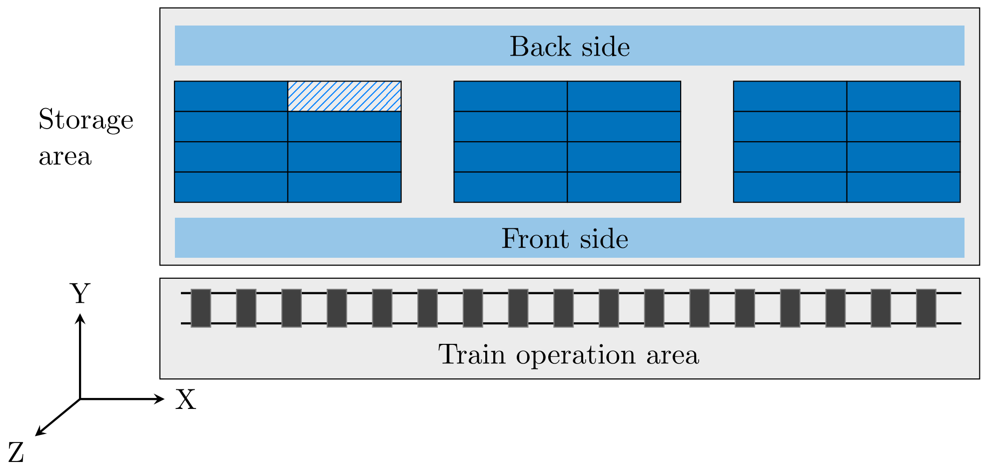

Figure 1: Aerial view of the considered container terminal layout

The storage area is divided into several parts according to the containers' destinations. As shown in Figure 1, lots are located along the track. Each lot consists of several stacks with a given maximum height. The coordinates X, Y and Z indicate the exact position of each container in the storage area. The X-coordinate refers to the lot. The Y-coordinate indicates the depth and the Z-coordinate specifies the vertical position of a container. Depending on the handling equipment, the storage area can be accessed from above, from the front side of the stack (seen from the track), or from the back side of the stack (as it is appropriate for the lined container in Figure 1).

When a load unit is due to be loaded on the train, a crane retrieves the container from the storage area, carries it to the train, and loads it onto its assigned slot. If the container is not directly accessible, it is necessary to rehandle blocking containers. The processes for unloading the trains are out of scope of this paper.

3.2. Containers

The load units considered in this paper are standardized containers. As we consider a static problem, the availability of containers or trains does not change during the loading process. Each container is characterized by its size (in the North American market 20 ft, 40 ft, 45 ft, 48 ft and 53 ft), its height (low-cube or high-cube containers), its weight, 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.

3.3. Intermodal double-stack trains

Railcars that have a common destination and leave the terminal in the same train are called a *block* and trains consist of several blocks. In practice, LPSPs are typically solved for blocks and not for the whole train. A block consists of a given sequence of railcars. Each railcar is defined by its type listed in a catalogue widely used in practice (Association of American Railroads, 2017). The railcars consist of between one and five platforms. Platforms can be either single-stack or double-stack and are characterized by various technical features. Single-stack platforms have one slot, whereas double-stack platforms are equipped with a bottom slot and a top slot.

Railcars can be used in several configurations differing in the number and length of loaded containers. We refer to these configurations as loading patterns. 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). As an example, Table 2 gives a subset of loading patterns for a given railcar with one platform.

Container length [ft] Pattern	20	40	45	48	53
1	1	0	0	0	0
2	2	0	0	0	0
3	0	1	0	0	0
4	2	1	0	0	0
5	2	0	1	0	0
6	2	0	0	1	0
7	2	0	0	0	1
8	0	2	0	0	0
9	0	1	1	0	0
10	0	1	0	1	0
11	0	1	0	0	1

Table 2: Example of a set of loading patterns for a one-platform railcar: each cell corresponds to the number of containers of a specific length that are included in a given pattern (Mantovani et al., 2018).

Besides the loading patterns, additional constraints related to the stability of the load units apply. 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 (Association of American Railroads, 2017; Mantovani et al., 2018).

We assume that all railcars of the block are empty at the start of the loading process. A top slot can only be loaded if either two 20 ft containers or one container measuring at least 40 ft has been loaded before on the bottom slot of the same platform.

3.4. Handling equipment

Terminals are equipped with special cranes – we consider gantry cranes and reach stackers – that handle the intermodal cargo within the terminal. Gantry cranes are immobile facilities that pick a container from above. Reach stackers, by contrast, are vehicles that lift containers from the side. In this paper, we assume that only one gantry crane or one reach stacker is loading a given block for which we solve the LPSP. The schematic designs of both gantry cranes and reach stackers are depicted in Figure 2.

(a) A gantry crane (Adobe Stock, 2018a)

(b) A reach stacker (Adobe Stock, 2018b)

Figure 2: Schematic design of gantry cranes and reach stackers

Obviously, not every container can be retrieved by the handling equipment at any time. Therefore, we define accessibility rules for each container. For a gantry crane the rules are simple: a container can only be retrieved if it is the uppermost container of its stack.

For reach stackers, however, the rules are more complex. Reach stackers retrieve containers either from the front side or from the back side of the lot (cf. Figure 1). 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.

Three sets of forbidden loading sequences are defined: one for the gantry crane (M^G), one for the reach stacker retrieving the containers from the front side (M^{RF}), and one for the reach stacker from the back side (M^{RB}). The rules are either derived by geometric dependencies or by 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 row: $X^i = X^{i'}$, $Y^{i'} + 1 = Y^i$, $g_i > \theta^1$

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

(f) – (i) analogously to (b) - (e) restrictions for reach stackers 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 movements are illustrated in Figure 3 and can easily be adapted to local restrictions.

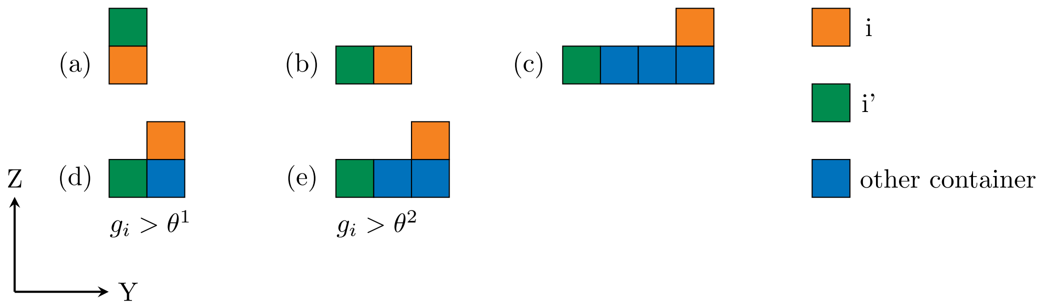

Figure 3: Overview of forbidden loading sequence $i \rightarrow i'$ (for gantry cranes: (a), for reach stackers from the left hand side (a) – (e))

If a forbidden sequence is executed, a container is inaccessible. It is then necessary to either retrieve the container from the other side of the storage area (if possible) or to move at least one blocking container apart, such that the required container can be reached. The blocking containers may stay in the terminal or be loaded to the train at a later moment and must then be touched again. We refer to this procedure as *double touching* or *rehandling* (in the literature, it is also known as *reshuffling*).

3.5. Challenges and objective

The LPSP for double-stack intermodal trains aims at assigning stored containers to slots on a train and at finding an optimal sequence of loadings such that the value of unloaded containers, the setup costs of the train, and the handling costs in the terminal are minimized. The handling costs comprise the distance covered by the crane, the number of rehandled containers, and the number of detours to the back side made by the crane. As illustrated in Figure 1, the reach stacker covers a distance that can be appropriately expressed by taxicab geometry, which is used in the objective function.

The considered problem comprises numerous interdependent decisions. The decision on the loading pattern defines how many containers of each size can be loaded to each platform. The assignment of containers to slots is made by respecting the loading patterns and additional constraints such as weight restrictions. This assignment imposes restrictions on feasible loading sequences as containers assigned to top slots can only be loaded if the bottom slot is full. The loading sequence gives us information about which containers cause a detour or force other containers to be rehandled due to inaccessibility. Thus, inopportune container-slot assignments can be derived. Changes in the load plan may affect feasible load sequences again or reduce the handling effort.

Summarizing, the LPSP for double-stack intermodal trains is defined as follows: Given a set of containers stored in a terminal with their characteristics and position, a sequence 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 unloaded containers and the handling cost is minimized.

4. ILP formulations and solution procedure

In this section, we introduce several formulations for the LPSP for double-stack intermodal trains. All formulations are based on the ILP for the load planning problem proposed by Mantovani et al. (2018) that we describe in Section 4.1. In Section 4.2, we introduce sets, notation, and decision variables that are used in all sequencing formulations. Next, in Sections 4.3 – 4.8, we define six different ILP formulations. Table 3 gives an overview of the formulations. They differ in the number of constraints and variables that they contain, as well as in the strength of their LP relaxation. The formulations can most easily be distinguished by the meaning of the variables. The sequencing variables in the formulations A1, B1, and C1 are process-oriented: they indicate the stage at which a container loading occurs (loaded *at* stage t). By contrast, the variables of the remaining formulations are state-oriented: they refer to the fact that a container has been loaded at or before a certain stage or not (loaded *by* stage t). The latter simplifies the writing of some constraints at the cost of additional constraints. The A formulations contain three-index sequencing variables resulting in a high number of variables and a rather low number of constraints. The B formulations use two-index sequencing variables which carry less information. This results in fewer variables but considerably more constraints. The C formulations are between the A and B ones in the sense that they complement the set of sequencing variables introduced in the B formulations by an additional set of variables related to the fullness state of each bottom slot. In Section 4.9, we describe the applied solution procedure.

Variables	z_{iqt}		z_{it}		z_{it}, b_{qt}	
	Name	Section	Name	Section	Name	Section
Process-oriented	A1	4.3	B1	4.5	C1	4.7
State-oriented	A2	4.4	B2	4.6	C2	4.8

Table 3: Names of the presented LPSP formulations

4.1. Formulation of the load planning problem

As the the model proposed by Mantovani et al. (2018) is the only one that deals with a high variety of loading patterns for double-stack railcars, we use this formulation as a basis for the extended problem including the loading sequence. The scope of the load planning model is to assign the containers $i \in N$ to the slots $q \in Q$ of a block in an intermodal terminal. Each railcar $j \in J$ can be loaded according to given loading patterns $k \in K_j$.

The railcars are characterized by their setup costs τ_j and their set of loading patterns K_j. For each platform, information is given on the length of its bottom slot L_p, and the maximum weight-carrying capacity G_p. The parameter μ_q is 1 if slot q is a bottom slot, and 0 otherwise. We denote by Q_μ the set of all bottom slots and by Q_p the slots of platform p. Each container is defined by its commercial value π_i, its length l_i, and its weight g_i. Containers are either part of the subset of low-cube N^{LC} or high-cube N^{HC} containers. 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.

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. The decision variables v_{iq} take care of the container-slot assignments. Finally, the auxiliary variables y_{ip} and x_{ij} assign containers to platforms and to railcars, respectively. The load planning problem can be written as follows:

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

s.t.

$$\sum_{q \in Q} v_{iq} \leq 1 \qquad \forall i \in N \tag{2}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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: k_i is a constant that takes value 0.5 for containers of length 20 ft and 1 for all other containers. 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. Finally, the center of gravity constraints are formulated in (10) for low-cube containers and in (11) for high-cube containers, respectively.

The presented formulation assigns a loading pattern to each railcar, and a container to a slot on a railcar. In the original model, the assignment of a container to a slot on a given platform can be determined or changed in a post-processing step. As our aim is to define the load planning and sequencing simultaneously, we add the constraints (3) ensuring a feasible slot assignment. We refer to Mantovani et al. (2018) for a more in-depth explanation of the model and for numerical results.

4.2. Notation and variables for all LPSP formulations

First, we introduce the set T of all stages during the loading process. We allow one container loading per stage and refer to the first stage as t^f and to the last one as t^l. In order to reduce symmetry, we ensure that all loadings are contiguous at the beginning of the time horizon. For the objective function, we need three cost parameters: β^1_{iq} represents the distance cost between the stacking position of container i and slot q related to the taxicab distance, β^2 is the cost for the rehandling of a container, and β^3 represents the cost for a detour of the reach stacker to the back side of the lot.

For all formulations below, we define three sets of binary variables: d_i taking value 1 if container i is double touched, γ_i taking value 1 if the reach stacker retrieves container i from the back side of the lot, and u_{qi} taking value 1 if the handling equipment reverses from slot q to the stacking position of 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 mentioned above, the movements of the reach stacker are more complex than those of the gantry crane. Therefore, we explicitly refer to the reach stacker movements in the formulations. For the gantry crane movements, 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. The constraints concerning the movements M^{RB} are unnecessary for this case.

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)

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

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

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

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

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

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

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

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

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, e.g., 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}. 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. If i' is rehandled or i is reached from the other side of the lot, the forbidden sequence is bypassed. Constraints (26) ensure that the top slot of each platform is loaded only if the bottom slot has been filled.

4.4. Formulation A2

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

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

s.t.

Constraints (2) – (18)

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

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

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

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

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

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

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

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

$$z_{iqt} \in \{0,\, 1\} \qquad \forall i \in N,\, \forall q \in Q,\, \forall t \in T. \tag{37}$$

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 large number of additional constraints (30).

4.5. Formulation B1

As the A formulations involve a large number of sequencing variables (i.e., $|N| \cdot |Q| \cdot |T|$), 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 train in stage t. The slot to which a container is assigned can be obtained from the v_{iq} variables (13) of the load planning problem. This leads to a reduction in the number of z_{it} variables by a factor $|Q|$ and results in the following formulation:

$$\min \sum_{i\in N} \pi_i \left(1 - \sum_{q\in Q} v_{iq}\right) + \sum_{j\in J} \tau_j \sum_{k\in K_j} w_{jk} + \sum_{i\in N}\left(\sum_{q\in Q} \beta^1_{iq}\left(v_{iq} + u_{qi}\right) + \beta^2 d_i + \beta^3 \gamma_i\right) \tag{38}$$

s.t.

Constraints (2) – (18)

$$\sum_{t\in T} z_{it} = \sum_{q\in Q} v_{iq} \qquad \forall i \in N \tag{39}$$

$$\sum_{i\in N} z_{it} \leq 1 \qquad \forall t \in T \tag{40}$$

$$\sum_{i\in N}\left(z_{i(t+1)} - z_{it}\right) \leq \sum_{i\in N}\left(z_{it} - z_{i(t-1)}\right) \qquad \forall t \in T \setminus \{t^f, t^l\} \tag{41}$$

$$z_{it} + v_{iq} + z_{i'(t+1)} - u_{qi'} \leq 2 \qquad \forall i \in N,\, \forall i' \in N,\, \forall q \in Q,\; \forall t \in T \setminus \{t^l\} \tag{42}$$

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

$$\sum_{\ell=0}^{t} z_{i\ell} \leq \sum_{\ell=0}^{t} z_{i'\ell} + d_{i'} - \gamma_i + 1 \qquad \forall (i,\, i') \in M^{RB},\, \forall t \in T \tag{44}$$

$$\sum_{\ell=0}^{t} z_{i'\ell} \leq \sum_{\ell=0}^{t} z_{i\ell} - \sum_{q \in Q_p} \left(v_{iq}\mu_q + v_{i'q}\left(1-\mu_q\right)\right) + 2 \quad \forall p \in P, \forall i \in N, \forall i' \in N, \forall t \in T \quad (45)$$

$$z_{it} \in \{0, 1\} \quad \forall i \in N, \forall t \in T. \quad (46)$$

Constraints (39) make sure that each assigned container is loaded during the time horizon, whereas (40) limit the number of loadings per stage to 1. All stages without loading are shifted to the end (41). Constraints (42) ensure that the variables u_{qi} take value 1 if the handling equipment reverses from slot q to container i. Constraints (43) ensure that either i' is moved before i, i' is double touched or i is picked from the back side. Constraints (44) work equivalently for the containers that are retrieved from the back side. The right loading order of each platform is guaranteed by (45). Each container i' loaded in the top slot can only be loaded after container i has been loaded in the bottom slot first.

4.6. Formulation B2

Similar to the formulation A2 (Section 4.4), this formulation aims at simplifying the constraints (43) - (45) and therefore uses the decision variable z_{it} to indicate whether container i has been loaded by t. The model can be written as follows:

$$\min \sum_{i \in N} \pi_i \left(1 - \sum_{q \in Q} v_{iq}\right) + \sum_{j \in J} \tau_j \sum_{k \in K_j} w_{jk} + \sum_{i \in N} \left(\sum_{q \in Q} \beta^1_{iq}\left(v_{iq} + u_{qi}\right) + \beta^2 d_i + \beta^3 \gamma_i\right) \quad (47)$$

s.t.

Constraints (2) – (18)

$$z_{it^l} = \sum_{q \in Q} v_{iq} \quad \forall i \in N \quad (48)$$

$$z_{it} \leq z_{i(t+1)} \quad \forall i \in N, \forall t \in T \setminus \{t^l\} \quad (49)$$

$$\sum_{i \in N} z_{it} \leq \sum_{i \in N} z_{i(t-1)} + 1 \quad \forall t \in T \setminus \{t^f\} \quad (50)$$

$$\sum_{i \in N} \left(z_{i(t+1)} - z_{it}\right) \leq \sum_{i \in N} \left(z_{it} - z_{i(t-1)}\right) \quad \forall t \in T \setminus \{t^f, t^l\} \quad (51)$$

$$z_{it} - z_{i(t-1)} - z_{i't} + z_{i'(t+1)} + v_{iq} - u_{qi'} \leq 2 \quad \forall i \in N, \forall i' \in N, \forall q \in Q, \forall t \in T \setminus \{t^f, t^l\} \quad (52)$$

$$z_{it} \leq z_{i't} + d_{i'} + \gamma_i \quad \forall (i, i') \in M^{RF}, \forall t \in T \quad (53)$$

$$z_{it} \leq z_{i't} + d_{i'} - \gamma_i + 1 \quad \forall (i, i') \in M^{RB}, \forall t \in T \quad (54)$$

$$z_{i't} \leq z_{it} - \sum_{q \in Q_p} \left(v_{iq}\mu_q + v_{i'q}\left(1-\mu_q\right)\right) + 2 \quad \forall p \in P, \forall i \in N, \forall i' \in N, \forall t \in T \quad (55)$$

$$z_{it} \in \{0, 1\} \quad \forall i \in N, \forall t \in T. \quad (56)$$

Constraints (48) ensure that each assigned container has been loaded in the last stage. Constraints (49) ensure that variables z_{it} stay at value 1 once i has been loaded, whereas constraints (50) limit the number of loadings per stage to 1. Constraints (51) forbid stages without loading between two with loadings. Constraints (52) make sure that the variables v_{iq},

z_{it}, and u_{qi} are correctly synchronized. Constraints (53) and (54) take care of the accessibility of the containers in the stack and constraints (55) guarantee the right order of loading with respect to the bottom and the top slot of each platform.

4.7. Formulation C1

As it can be seen in the constraints (45) and (55), the correct loading of containers on double-stack railcars with respect to the order of bottom and top slot requires a large number of constraints in the B formulations. This is related to the lack of information as to whether bottom slot q is loaded at stage t. We now aim at reducing the number of these constraints significantly by introducing the auxiliary variable b_{qt} indicating whether bottom slot q is fully loaded at stage t. As in B1, the sequencing variables are activated if container i is loaded at stage t. A large part of formulation B1 can be used. The model can be stated as follows:

$$\min \sum_{i \in N} \pi_i \left(1 - \sum_{q \in Q} v_{iq} \right) + \sum_{j \in J} \tau_j \sum_{k \in K_j} w_{jk} + \sum_{i \in N} \left(\sum_{q \in Q} \beta^1_{iq} \left(v_{iq} + u_{qi} \right) + \beta^2 d_i + \beta^3 \gamma_i \right) \quad (57)$$

s.t.

Constraints (2) – (18) and (39) – (44)

$$\sum_{\ell=0}^{t} z_{i\ell} + \sum_{q \in Q_p} v_{iq} \left(1 - \mu_q \right) \le \sum_{q \in Q_p \cap Q_\mu} \sum_{\ell=0}^{t} b_{q\ell} + 1 \qquad \forall p \in P,\, \forall i \in N,\, \forall t \in T \quad (58)$$

$$b_{qt} \le \sum_{\ell=0}^{t} z_{i\ell} - v_{iq} + 1 \qquad \forall i \in N,\, \forall q \in Q_\mu,\, \forall t \in T \quad (59)$$

$$z_{it} \in \{0,\, 1\} \qquad \forall i \in N,\, \forall t \in T \quad (60)$$

$$b_{qt} \in \{0,\, 1\} \qquad \forall q \in Q_\mu,\, \forall t \in T. \quad (61)$$

Constraints (58) 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. Compared to (45), the number of constraints can be reduced by a factor of the number of containers ($|N|$). The additional constraints (59) 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.

4.8. Formulation C2

This formulation is similar to the formulation B2 and defines the sequencing variable z_{it} to be 1 if i has been loaded by stage t. Analogously to the formulation C1, 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:

$$\min \sum_{i \in N} \pi_i \left(1 - \sum_{q \in Q} v_{iq} \right) + \sum_{j \in J} \tau_j \sum_{k \in K_j} w_{jk} + \sum_{i \in N} \left(\sum_{q \in Q} \beta^1_{iq} \left(v_{iq} + u_{qi} \right) + \beta^2 d_i + \beta^3 \gamma_i \right) \quad (62)$$

s.t.

Constraints (2) – (18) and (48) – (54)

$$z_{it} \leq \sum_{q \in Q_p} \left(\mu_q b_{qt} - v_{iq}\left(1 - \mu_q\right)\right) + 1 \qquad \forall p \in P,\, \forall i \in N,\, \forall t \in T \tag{63}$$

$$b_{qt} \leq z_{it} - v_{iq} + 1 \qquad \forall i \in N,\, \forall q \in Q_\mu,\, \forall t \in T \tag{64}$$

$$z_{it} \in \{0,\, 1\} \qquad \forall i \in N,\, \forall t \in T \tag{65}$$

$$b_{qt} \in \{0,\, 1\} \qquad \forall q \in Q_\mu,\, \forall t \in T. \tag{66}$$

Constraints (63) make sure that the top slot is only filled if the bottom slot is full and constraints (64) synchronize the b_{qt} with the z_{it} variables.

4.9. 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. All reported instances are solved using the following procedure:

1. Solve the load planning problem described by Mantovani et al. (2018) to optimality using a 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 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 and the additional costs to the objective function.
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 from backside ;
  else
    load random container from front side whose assigned slot is ready and rehandle all blocking containers ;
  end
  update state of slots ;
end

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. The test instances are described in Section 5.1. In a first step, we run experiments with the formulations (Sections 4.3 - 4.8) considering full distance cost even though we expect this problem to be intractable for realistic size instances. Second, in Section 5.3, we report results with formulations considering a simplified distance cost motivated by the fact that one-way moves strongly correlate with the cost of the overall loading effort (Boysen et al., 2010). 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 see how the best performing formulations deal with larger instances. Last (Section 5.6), we report a case study highlighting a large reduction of the handling cost when applying the integrated model compared to sequential approaches. For all experiments, we compare the results for two types of handling equipment: a gantry crane and a reach stacker.

5.1. Test instances and setup

Since LSPSs are typically solved for blocks (cf. Section 3.3), we consider test instances related to the length of one block. Our basic test instances comprise 50 containers and a minimum block length of 667 ft. They are truncated versions of the instances described by Mantovani et al. (2018). We use five different sets of railcars. They are all part of the simple random samplings, which – compared to stratified random samplings – result in a higher share of railcars with high flexibility in terms of loading patterns (Mantovani et al., 2018). There are two types of container sets: five sets consider containers of equal length only (40 ft), the other five sets enclose containers of different lengths (40 ft, 45 ft, 48 ft, and 53 ft). The weights of the containers are randomly drawn from a normal distribution. Combining five railcar sequences with ten container sets yields a total of 50 instances.

We consider a layout of the storage area that is organized as follows (cf. Figure 1): the containers are stacked at a maximum height of three containers. There are six lots in total, which are arranged alongside the track. Depending on the number of containers, the lots vary in depth (for 50 containers, the depth is 3).

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 is of course beyond an acceptable amount of time for an operational problem. 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

We first test the formulations with full consideration of distance cost (Sections 4.3 - 4.8). The instances described in Section 5.1 are intractable for the full formulations and we thus truncate them in such a way that there remain 15 containers per instance. This corresponds to a block length of 200 ft.

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 those 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 gantry crane instances. For the instances considering reach stacker movements, a small share can be solved to optimality. The average optimality gap is rather large (44.2 %) even after ten hours of computation time. Since in 591 out of 600 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	C1	C2
			Gantry crane			
# opt. solved instances	0	0	0	0	0	0
Avg. opt. gap [%]	43.6	44.1	43.8	43.9	44.0	43.7
			Reach stacker			
# opt. solved instances	1	1	4	3	7	5
Avg. CPU [s]	333	590	477	733	553	683
Avg. opt. gap [%]	44.5	46.9	44.9	44.6	42.6	43.8

Table 4: Computational results for 50 instances with block length of 200 ft (15 containers, full distance)

These computational results point out that even for very small academic 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 by neglecting crane movements without containers. As shown by Boysen et al. (2010), the cost for the loaded one-way moves strongly correlates with the cost of the overall effort, thus making this simplification reasonable.

5.3. Experiments with one-way distance cost

We now test the formulations with simplified one-way distance cost. We fix the value of the variables u_{qi} to 1 and remove these variables from the objective function. Accordingly, the constraints (23), (33), (42), and (52) are obsolete.

First, we solve the truncated instances of Section 5.2. Figure 4 displays the average number of constraints and variables for all formulations. The number of constraints can be drastically reduced by simplifying the distance cost.

The numerical results are reported in Table 5. All instances can now be solved to optimality within short computational time (on average 11 seconds for gantry crane and 26 seconds for reach stacker movements). Three out of 50 instances require one rehandled container for reach stacker movements, one of them additionally causes a detour for the crane. All other instances are solved without any unproductive movement.

Formulation	A1	A2	B1	B2	C1	C2
			Gantry crane			
# opt. solved instances	50	50	50	50	50	50
Avg. CPU [s]	3	31	10	15	6	4
			Reach stacker			
# opt. solved instances	50	50	50	50	50	50
Avg. CPU [s]	18	86	21	20	6	7

Table 5: Computational results for the 50 instances of block length 200 ft (15 containers, one-way distance)

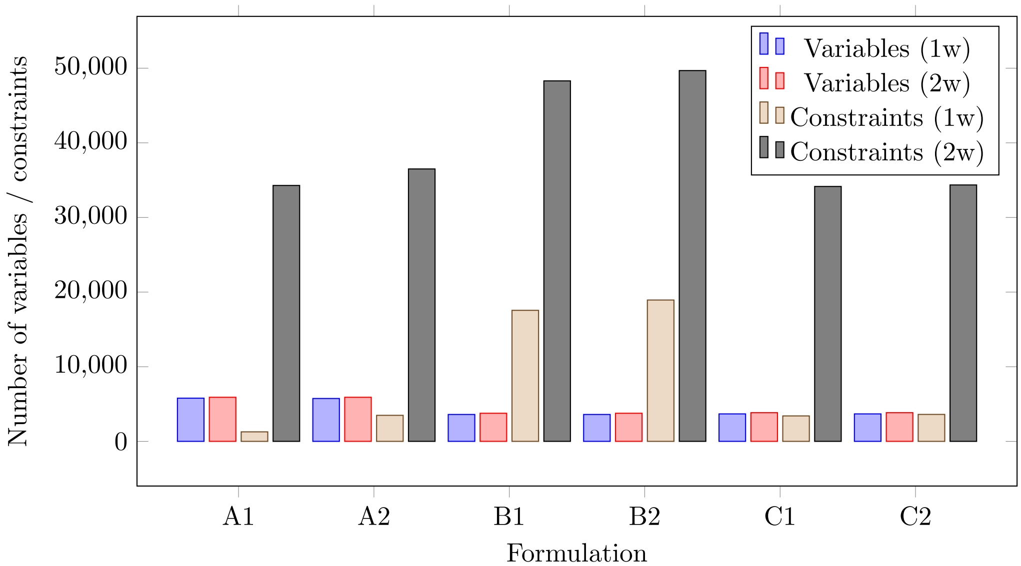

Figure 4: Average number of variables and constraints per formulation for the 200 ft instances including gantry crane and reach stacker movements with consideration of (2w) two-way and (1w) one-way distance cost

These convincing computational results motivate us to continue with the simplified distance cost and to apply it on the 667 ft instances. To give an idea about the different sizes of the mathematical models, we depict the average number of variables and constraints in Figure 5 (logarithmic scale of y-axis). The A formulations have the largest number of variables (on average 86,182), followed by the C formulations (14,400) and the B formulations (13,565). The difference in the number of variables related to the handling equipment is quite small. The number of constraints is lowest for the formulation A1 (6,077 for gantry crane, and 20,817 for reach stacker) and ranges between 78,638 and 95,658 for the formulations A2, C1, and C2. Due to the constraints (45) and (55), the number of constraints for the B formulations is much higher (between 1.82 M and 1.89 M). Comparing the reach stacker to the gantry crane movements, the number of constraints increases by roughly 15,000 on average.

The numerical results for the optimization are given in Table 6. For a better interpretation of the optimality gap, the average number of rehandlings and retrievals from the back side of the lot are given for instances that could not be solved to optimality. The results show that the problems with reach stacker movements are harder to solve than those with gantry crane movements: the average number of optimally solved instances drops from 38 to 28, the average computational time rises from roughly two hours to four hours and the average optimality gap increases from 9 % to 25 %. This is related to the higher number of forbidden loading sequences. For both cranes, all optimal solutions go without unproductive movements.

For the gantry crane, the number of instances solved to optimality varies between 26 and 43 out of a total of 50. The formulation C1 finds the highest number of optimal solutions, followed by formulations A1 and C2. Concerning the computational times, the C formulations clearly outperform the others. Regarding the optimality gaps, the formulations C1, C2, and A2 evidently dominate the others. The average number of rehandled containers for instances that are not optimally solved at timeout is rather low for all formulations.

The instances containing reach stacker movements, however, show a different picture. Whereas the number of optimally solved instances decreases significantly for the A and B

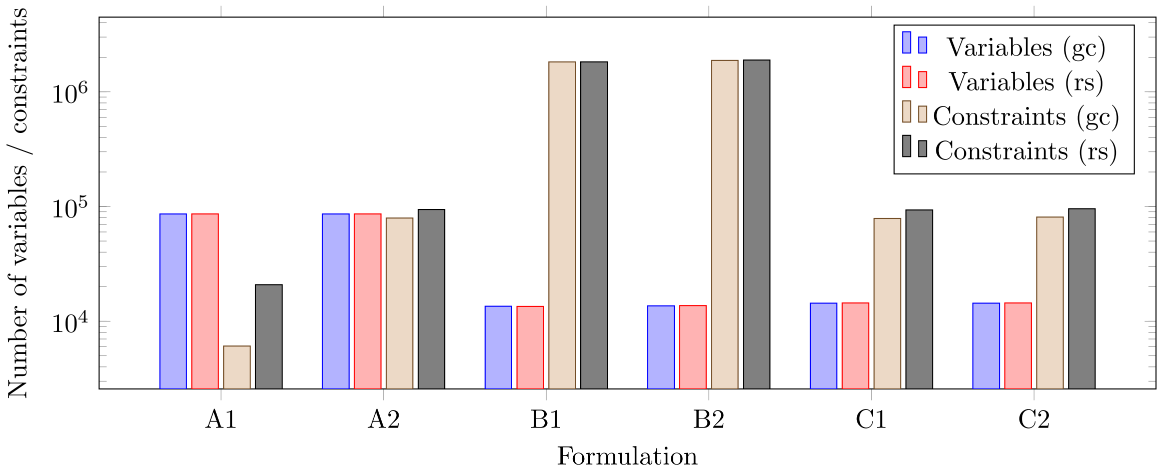

Figure 5: Average number of variables and constraints per formulation for the 667 ft instances (one-way distance, gc: gantry crane, rs: reach stacker)

Formulation	A1	A2	B1	B2	C1	C2
			Gantry crane			
# opt. solved instances	42	37	26	38	43	42
Avg. CPU [s]	5,418	9,268	20,526	12,446	1,423	821
Avg. opt. gap [%]	25.5	5.0	14.6	7.0	0.6	0.6
Avg. # rehandlings (timeout)	1.5	0.2	1.4	0.3	0.0	0.0
			Reach stacker			
# opt. solved instances	26	22	9	29	41	40
Avg. CPU [s]	16,897	19,462	30,401	17,181	1,388	1,041
Avg. opt. gap [%]	61.4	7.7	44.1	34.5	0.6	0.6
Avg. # rehandlings (timeout)	4.5	0.1	4.9	1.8	0.0	0.0
Avg. # detours (timeout)	2.4	0.2	2.0	0.9	0.0	0.0

Table 6: Computational results for the 50 instances of block length 667 ft (50 containers, one-way distance)

formulations, the numbers remain almost stable for the C 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, C1, and C2.

We now examine whether and how the container characteristics have an impact on the computational results. Figure 6 compares the number of optimally solved instances for two settings of container characteristics. Interestingly, the numbers differ a lot. On average 15 (resp. 12) out of 25 instances can be solved to optimality using a gantry crane (resp. reach stacker) considering instances with containers of equal length. The share of solved instances with containers of four different lengths is considerably higher (23 for gantry crane, 16 for reach stacker). Due to the symmetry of load plans, the solution space for containers of equal size is larger.

In summary, the large number of constraints of the B formulations slows the model down significantly. The C formulations dominate the other formulations in terms of consistency, computational time, and optimality gap. The reduction in the number of variables (com-

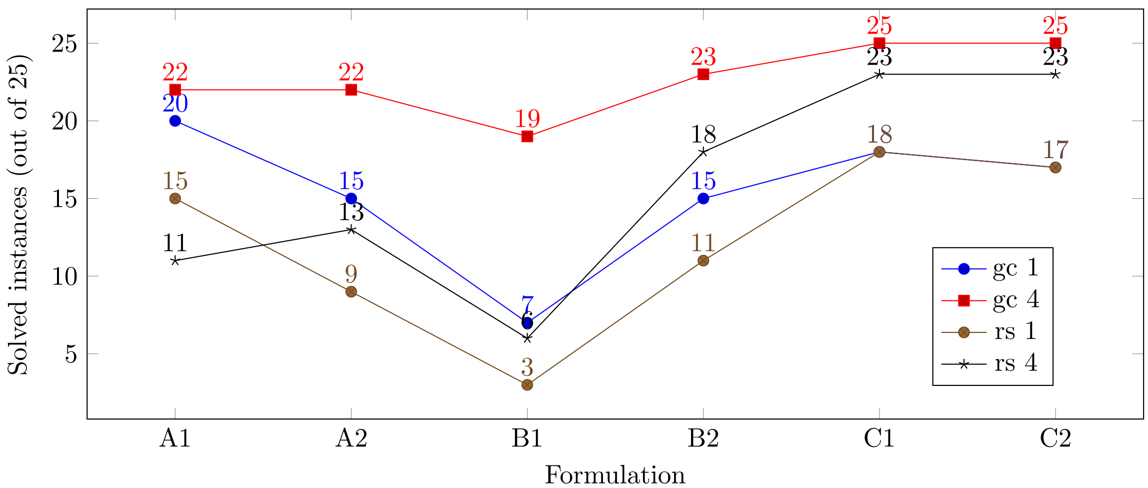

Figure 6: Number of solved instances with respect to container characteristics (1 size: 40 ft, 4 sizes: 40 ft, 45 ft, 48 ft, and 53 ft), handling equipment (gc: gantry crane, rs: reach stacker) and formulation

pared to A) 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 reach stacker movements. On average, the C formulations show reasonable computing times below 1,800 seconds. Except for the B formulations the two different definitions of the z variables (i loaded *at* or *by* t) seem not to affect the performance strongly. For the B formulations, the second definition works considerably better. With respect to the solution quality at timeout, A2 (resp. B2) dominates A1 (resp. B1).

5.4. Experiments ignoring distance cost

Many intermodal terminals are built in urban areas where space is a scarce resource. Thus, their dimensions are rather small and the distance covered by the handling equipment while loading the train may be negligible. In this context, the relevance of the particular term related to the distance in the objective function is questionable. In this section, we remove the term from the objective function and use (67) to investigate the changes in the solution process:

$$\min \sum_{i \in N} \pi_i \left(1 - \sum_{q \in Q} v_{iq} \right) + \sum_{j \in J} \tau_j \sum_{k \in K_j} w_{jk} + \sum_{i \in N} \left(\beta^2 d_i + \beta^3 \gamma_i \right). \tag{67}$$

The numerical results for these experiments are reported in Table 7. Similarly to the results presented in Section 5.3, the reach stacker instances are harder to solve. On average, 47 (resp. 38) out of 50 instances considering a gantry crane (resp. reach stacker) as handling equipment could be solved to optimality. Compared to the results with one-way distance cost (38 and 28 instances), these figures are considerably higher.

For the gantry crane, all problems can be solved to optimality with the A and C formulations. The average computation time drops from roughly 94 (resp. 157) 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 C1 and C2 are on average 11 and 17 minutes, respectively. The number of instances solved increases for the formulations B1 and B2 to 42

Formulation	A1	A2	B1	B2	C1	C2
			Gantry crane			
# opt. solved instances	50	50	42	40	50	50
Avg. CPU [s]	1,091	3,969	17,945	14,755	651	995
Avg. opt. gap [%]	-	-	97.8	97.9	-	-
Avg. # rehandlings (timeout)	-	-	5.9	6.6	-	-
			Reach stacker			
# opt. solved instances	48	47	14	18	50	50
Avg. CPU [s]	8,935	11,546	29,145	24,025	1,210	1,361
Avg. opt. gap [%]	100.0	95.5	98.3	98.8	-	-
Avg. # rehandlings (timeout)	16.5	1	14.4	9.9	-	-
Avg. # detours (timeout)	10	1	4.7	4.2	-	-

Table 7: Computational results for the 50 instances of block length 667 ft (50 containers, no distance)

and 40. The computational times are on average much higher than for the other formulations (299 and 246 minutes).

The reach stacker movements intensify the differences between the formulations. The C formulations are the only ones for which one can solve all 50 instances to optimality. The average computation time is 20 minutes for C1 and 23 minutes for C2, respectively. The formulations A1 and A2 solve 48 and 47 instances. The computational times are 149 and 192 minutes, and hence much higher compared to C1 and C2. The formulations B1 and B2 are only able to solve 14 and 18 instances in more than eight and six hours on average.

For all formulations, the optimality gap for instances without proven optimum is very large ($> 95\%$). The number of unproductive movements at timeout is reasonable for the instances solved by formulation A2. In terms of computational time, again, the C formulations clearly outperform the others. The A formulations can still deal with the instance size, whereas the B formulations do not perform reliably on these instances.

The figures show that the costs for the distance in the objective function makes the optimization problem considerably harder. Therefore, it is expedient to discard this term if the layout of the terminal is rather compact and distance costs are negligible.

5.5. Experiments with larger instances

The computational experiments reported in Sections 5.3 and 5.4 indicate that the formulations C1 and C2 work best with respect to reliability and computational time. We now use these formulations without consideration of the distance on larger instances to examine which instance size can be solved in reasonable time. The results are reported in Table 8 and Figure 7. Recall that the largest instances that solved to optimality for the LPSP for single-stack trains in a similar setting comprise 40 containers (Ambrosino et al., 2013).

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 instances comprising 1,000 ft (75 containers) can be solved to optimality for the gantry crane movements. For the reach stacker instances, however, only 28 of the instances can be solved to optimality for the C1 and 36 for the C2 formulations. The computational times are above five hours and thus not appropriate for operational problems.

Formulation	C1	C2	C1	C2	C1	C2	C1	C2
Instance size	667 ft		1,000 ft		1,500 ft		2,000 ft	
# Containers	50		75		113		150	
				Gantry crane				
# 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
				Reach stacker				
# 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 8: Computational results for the large instances (50 of each size) without consideration of distance cost

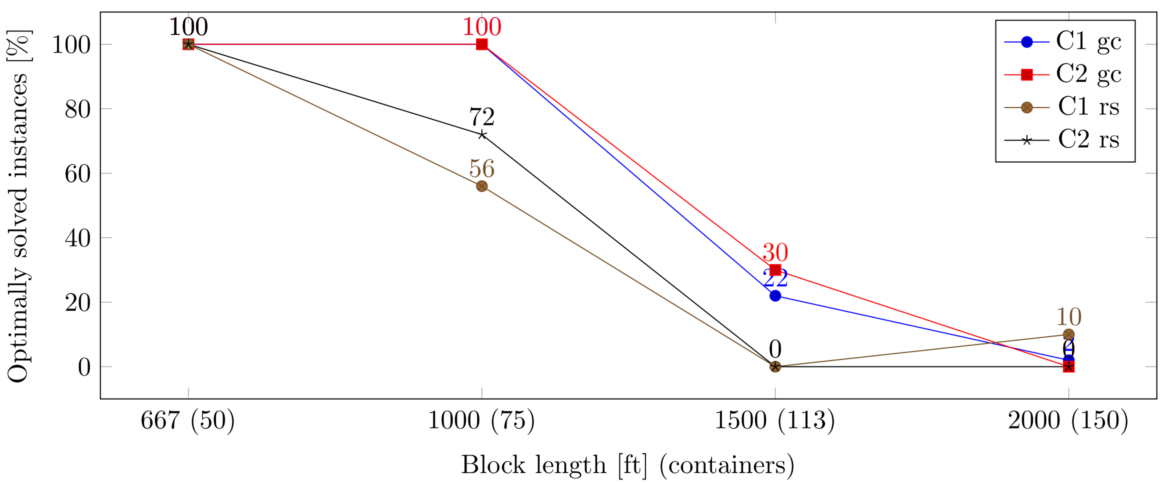

Figure 7: Share of solved instances (out of 50) with respect to instance size, formulation, and handling equipment (no distance considered, gc: gantry crane, rs: reach stacker).

Bigger instances comprising a block of at least 1,500 ft length and 113 containers can hardly be solved considering the gantry crane. Only 11 (resp. 15) instances are solved optimally with the formulation C1 (resp. C2) within the given time limit. None of the instances with reach stacker movements can be solved to optimality. Note that the average optimality gap by the end of the computational time is very large (> 95 %). The solution quality in terms of unproductive movements at timeout is poor except for the C2 formulation with instance size of 1,000 ft.

The largest instances that we test comprise 2,000 ft and 150 containers. Only very few instances can be solved to optimality. The surprinsingly fast computational times are related to the fact that in all six cases, 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 667 ft 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. As not all instances can be solved to optimality with the integrated model (Table 6), we only compare instances that are optimally solved with at least one of the six formulations (49 instances for a gantry crane, 41 instances for a reach stacker). The distribution of the number of rehandled containers obtained by the sequential model is displayed in Figure 8.

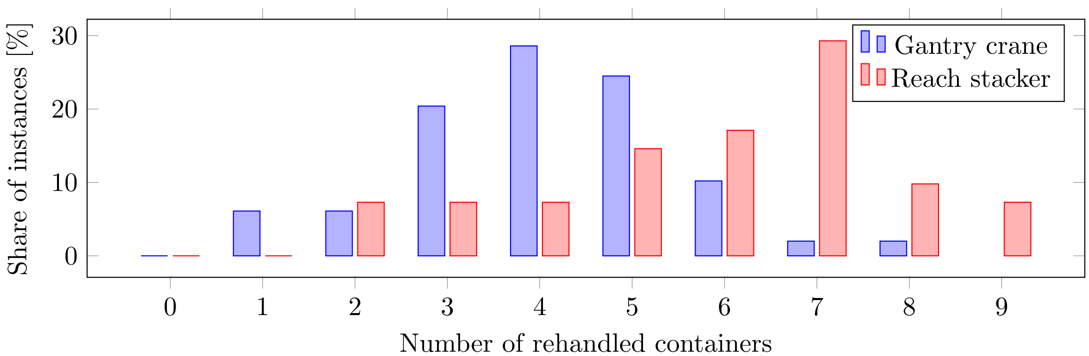

Figure 8: Distribution of the number of rehandled containers for the sequential solving of the LPSP

For the gantry crane experiments, up to 8 containers (on 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 reach stacker instances, the number of double touched containers varies between 2 and 9 (on average 5.9) and drops to 0. The number of detours varying between 0 and 2 (on 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. 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 unloaded 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.

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 flexible way 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 six different ILP formulations.

Computational results show that the instances considering reach stackers instead of gantry cranes are much 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 two out of six formulations (C1 and C2) outperform the others. 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 seem to be much less affected by the higher number of forbidden sequences caused by the reach stacker.

Additionally, we show that by ignoring the costs for the distances occurring in the terminal, as it may be suitable for terminals with a compact layout, we can solve instances with a block length of 667 ft and 50 containers with a commercial solver in reasonable time. 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 highlight 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. This confirms that an integrated solution of the LPSP can be of great benefit for terminal operators.

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 double touches is a relevant extension of the model as, in rare cases, the rehandling of blocking containers may involve further rehandlings. Tailored solution methods 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.

References

Adobe Stock (2018a). Rail Mounted Gantry Crane. URL: https://stock.adobe.com/ca/images/rail-mounted-gantry-crane/105309446?prev_url=detail.

Adobe Stock (2018b). Reach Stacker with Container. URL: https://stock.adobe.com/ca/images/reach-stacker-with-container/104697806?prev_url=detail.

Aggoun, A., Rhiat, A., & Grassien, J.-P. (2011). Online Assignment of Container to Train Using Constraint Programming. In C. Sombattheera, A. Agarwal, S. K. Udgata, & K. Lavangnananda (Eds.), *Multi-disciplinary Trends in Artificial Intelligence* (pp. 395–405). 5th International Workshop, MIWAI 2011 Berlin Heidelberg: Springer-Verlag. doi:10.1007/978-3-642-25725-4_35.

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., & 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.

Dowsland, K. A., & Dowsland, W. B. (1992). Packing Problems. *European Journal of Operational Research*, *56*, 2–14. doi:10.1016/0377-2217(92)90288-K.

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.

Dyckhoff, H., Scheithauer, G., & Terno, J. (1997). Cutting and packing. In M. Dell'Amico, F. Maffioli, & S. Martello (Eds.), *Annotated Bibliographies in Combinatorial Optimization* chapter 22. (pp. 393–414). New York, NY, USA: John Wiley & Sons.

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.

Imai, A., Sasaki, K., Nishimura, E., & Papadimitriou, S. (2006). Multi-Objective Simultaneous Stowage and Load Planning for a Container Ship with Container Rehandle in Yard Stacks. *European Journal of Operational Research*, *171*, 373–389. doi:10.1016/j.ejor.2004.07.066.

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.

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.

Martello, S., & Toth, P. (1990). *Knapsack Problems: Algorithms and Computer Implementations*. New York, NY, USA: John Wiley & Sons, Inc.

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.

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-Verlag 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.

Vis, I. F. A., & de Koster, R. (2003). Transshipment of Containers at a Container Terminal: An Overview. *European Journal of Operational Research*, *147*, 1–16. doi:10.1016/S0377-2217(02)00293-X.

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.

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.