Transportation Research Part E 89 (2016) 70–85

Contents lists available at ScienceDirect

Transportation Research Part E

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

# A model for a multi-size inland container transportation problem

Julia Funke *, Herbert Kopfer

*Chair of Logistics, University of Bremen, 28359 Bremen, Germany*

ARTICLE INFO

*Article history:*
Received 31 July 2015
Received in revised form 28 January 2016
Accepted 27 February 2016
Available online 11 March 2016

*Keywords:*
Container hinterland transportation
Mixed-integer programming
Multi-size containers
Pickup and delivery

ABSTRACT

In the multi-size Inland Container Transportation Problem (mICT) trucks are able to transport up to two 20-foot or one 40-foot container at a time along routes with various pickup and delivery locations. A mixed-integer linear program for the mICT is presented using two alternative objective functions: minimization of the total travel distance and minimization of the total operation time of the trucks. The presented model is tested on instances which vary in size. Computational experiments show that by means of the presented model small problem instances can be solved optimally.



## 1. Introduction

*Intermodal container transportation* refers to the movement of empty containers or containerized cargo by different means of transportation (modes) in one transportation chain. A typical transportation chain can be subdivided into three sections, each section being operated by a distinct transportation mode (Macharis and Bontekoning, 2004). The first section is called *pre-haulage*. In pre-haulage, trucks fetch containerized cargo from the actual customers (*senders*) by carrying fully loaded containers to terminals. The longest distances are covered in the *main-haulage*. This second section is mostly carried out by barge, deep sea shipping or rail to move containers between terminals. The last section implies container transportation by truck from terminals to customers (*receivers*) and is referred to as *end-haulage*. Between 25% and 40% of the total intermodal container transportation costs are accrued in the trucking sections (*drayage*) of an intermodal transportation chain (Macharis and Bontekoning, 2004). Notteboom and Rodrigue (2005) state that in sea transportation inland costs range between 40% and 80% of the total transportation costs and are thus even higher. Since most flows of containerized cargo are asymmetric, there is a need for unproductive movements caused by empty container repositioning. While the share of empty containers that are repositioned at sea is around 20% of all containers transported, the rate of repositioned empty containers on land is estimated to be even twice as high (Konings, 2005). Consequently, the movement of empty containers is a significant cost factor in container transportation which should not be neglected.

This paper addresses a problem arising in the field of drayage. One trucking company has to transport containerized cargo between customers, terminals and one depot. Empty containers needed as transportation media for cargo can be stored at the depot. All containers of a given size are assumed to be interchangeable with each other; i.e., for a cargo request it does not matter which container of that size is used to transport cargo. Then the problem is two-fold: container assignment and truck routing. On the one hand, empty containers have to be assigned to cargo transportation requests. There are two ways to

* Corresponding author.
*E-mail addresses:* jfunke@uni-bremen.de (J. Funke), kopfer@uni-bremen.de (H. Kopfer).

http://dx.doi.org/10.1016/j.tre.2016.02.010
1366-5545/

assign them: As the depot is assumed to be storage space for empty containers, empty containers which are assigned to cargo transportation requests can be obtained from and delivered to the depot. Another way to assign empty containers is constituted by the *street-turn* methodology. By performing a street-turn, empty containers are directly carried from receivers to senders (Jula et al., 2006). On the other hand, routes for trucks have to be constructed in such a way that the trucking company performs all container movements resulting from a given set of cargo transportation requests. The common problem formulation found in literature is restricted to the transportation of 40-foot containers only. Such intermodal drayage truck routing and scheduling problems are representatives of the class of Full Truckload Pickup and Delivery Problems (Erera and Smilowitz, 2008), or Full Truckload Pickup and Delivery Problems with Time Windows (FT-PDPTW), if in addition time restrictions are given. Surveys on pickup and delivery problems are given by e.g. Savelsbergh and Sol (1995) and Parragh et al. (2008a,b). The FT-PDPTW can be transformed into an asymmetric multiple Traveling Salesman Problem with Time Windows (amTSPTW). Among others Wang and Regan (2002), Jula et al. (2005), Zhang et al. (2010), and Braekers et al. (2013) show how to transform problems arising in the field of drayage into amTSPTW's. This paper extends the common problem formulation by considering containers of different sizes (i.e., 20-foot and 40-foot containers). Trucks can either transport up to two 20-foot containers or one 40-foot container at a time. To obtain a formal description of the considered problem, a graph representation and a mathematical formulation as a variant of the amTSPTW are presented. The model builds optimal routes that are allowed to consist of more than four stops for the transportation of 20-foot and 40-foot containers. As far as we know, the implementation of a mathematical model for the first time has been able to compute solutions to this problem type.

The remaining paper is organized as follows. A survey on research sources is presented in Section 2. In Section 3 a formal definition of the multi-size Inland Container Transportation Problem (mICT) is given. The mathematical model for the mICT presented in Section 4, and the results of computational experiments on randomly generated test instances as well as modified instances from literature sources are presented in Section 5. Section 6 concludes this paper.

## 2. Literature review

Although, the Traveling Salesman Problem (TSP) is NP-hard (Karp, 1972), there are fast methods for solving large instances optimally (see e.g. Hoffman et al., 2013 for a list of approaches). One of the exact methods is called *Concorde* (Applegate et al., 1995, 2011). Concorde is able to optimally solve instances like the 85,900-cities instance of *TSPLIB* (Reinelt, 1991). However, optimal solutions to instances of this size could only be obtained for the symmetric TSP. For some reasons, the asymmetric TSP seems to be much harder to solve (Grötschel, 2015). A generalization of the TSP is the multiple Traveling Salesman Problem (mTSP), which considers several salesmen. An overview on formulations and solution procedures for the mTSP is given by Bektas (2006). mTSP instances that could be optimally solved contain about 500 cities (Gavish and Srikanth, 1986). A further extension of the TSP/mTSP is to additionally consider time windows, in which salesmen have to arrive at cities. As Williams (2013) states, such a formulation could prove very difficult to solve for reasonable-sized problem instances. For surveys of time window constrained routing problems see e.g. Solomon (1987), Solomon and Desrosiers (1988), and Desroches et al. (1988).

Trucking problems arising in intermodal container transportation are for example studied by Jula et al. (2005). Jula et al. (2005) introduce a transportation problem, in which fully loaded 40-foot containers have to be transported between specified pickup and delivery locations respecting the given arrival time windows at the different locations. The authors model the problem as amTSPTW by introducing one node for each container pickup and delivery pair. They propose a two-phase algorithm based on dynamic programming that optimally solves the problem and additionally propose a hybrid methodology combining dynamic programming techniques with a genetic algorithm and compare the solutions constructed by this methodology with solutions of an insertion heuristic approach that is inspired by Jaw et al. (1986). If the simultaneous transport of fully loaded and empty containers is simultaneously considered, it is distinguished between *well-defined* and *flexible tasks* (Smilowitz, 2006). Well-defined tasks comprise the transportation of containers, in which pickup and delivery locations are known in advance. In contrast, either the point of origin or destination is left undefined for flexible tasks, and, thus, is a matter of optimization. Imai et al. (2007) study a problem including the following two kinds of flexible tasks: In delivery trips, a fully loaded container has to be carried from the intermodal terminal to a consignee, where the container is unloaded. The destination location of the obtained empty container can either be the intermodal terminal or it can be reused for a pickup trip. In a pickup trip, an empty container has to be transported either from a delivery location of a delivery trip or from the intermodal terminal to a shipper, where the container is filled. The obtained fully loaded container has to be transported to the intermodal terminal. In any case, i.e. for delivery and pickup trips, the assignment of a container to a truck remains unchanged during the whole trip from the truck's start at the terminal until its arrival at the terminal again. Consequently, new assignments of containers to trucks are only possible at the terminal. Imai et al. (2007) propose a subgradient heuristic approach based on Lagrangian relaxation to compute solutions to this problem definition. Caris and Janssens (2009) extend the problem definition of Imai et al. (2007) by introducing time windows and present a FT-PDPTW formulation. Afterward, Caris and Janssens (2009) improve an initial solution constructed by a two-phase insertion heuristic approach with local search techniques. By introducing multiple terminals, Zhang et al. (2010) extend the work of Zhang et al. (2009) and define the Inland Container Transportation Problem (ICT). The authors present a heuristic approach based on a window partitioning approach (Wang and Regan, 2002). Sterzik and Kopfer (2013) show a mathematical model and present a tabu search approach (Glover, 1986) to obtain solutions for the ICT. Nossack and Pesch (2013) model the ICT as FT-PDPTW. To solve instances of realistic size, Nossack and Pesch (2013) present a two-stage heuristic approach: In the first phase of the approach, flexible tasks are combined to well-defined tasks. The constructed well-defined tasks are assigned to and sequenced onto trucks. In the second phase, the initial solution that is built in the first phase is improved by an ejection chain heuristic approach (Pesch and Glover, 1997). Zhang et al. (2011) alter the ICT by restricting the number of containers stored at the depot and show that this modification substantially increases the complexity of the ICT. Braekers et al. (2013) study a problem that primarily differs from the ICT in enabling trucks to separate from containers and thus leave containers alone while they are unloaded; i.e., the assignment of containers to trucks may not only be changed at the terminal but at customer locations, too. Macharis and Bontekoning (2004) make a distinction between two different types of container (un-)loading strategies: In the *stay-with* procedure, trucks stay with containers during (un-)loading. In the *drop-and-pick* procedure, trucks and containers can separate, i.e., a fully loaded or empty container is dropped off at the customer's location. The truck can carry on with other tasks, while the container is (un-)loaded. Braekers et al. (2013) also allow the drop-and-pick procedure, formulate the problem as amTSPTW and propose a sequential and an integrated approach based on a deterministic annealing algorithm. Sterzik et al. (2012) show a further extension on the ICT. In this extension, containers cannot only be (un-) loaded using the drop-and-pick procedure, several companies are considered that can share empty containers between each other.

Several other papers deal with modifications of transportation problems of 40-foot containers, but only a few research papers consider 20-foot and 40-foot containers simultaneously. Chung et al. (2007) propose a heuristic approach based on an insertion heuristic approach (Rosenkrantz et al., 1977) to compute solutions to a problem considering well-defined tasks of 20-foot and 40-foot containers. In the problem definition of Chung et al. (2007), three types of vehicles are available: The first, respectively second type, is able to transport one 20-foot container, respectively one 40-foot container, while the third type of vehicles is able to transport one 40-foot container or two 20-foot containers at the most. Vidović et al. (2011) show a multiple matching integer program that is able to optimally solve small instances of a problem definition, in which empty and fully loaded 20-foot and 40-foot containers have to be transported by a homogeneous fleet of trucks having a capacity of two twenty-foot equivalent units (TEU). As Street-turns are excluded in the definition of Vidović et al. (2011), a single route contains at most four locations and the problem can be classified as a Vehicle Routing Problem with backhauls. For problems of larger sizes, Vidović et al. (2011) propose a heuristic approach based on matching utilities. Afterward, Popović et al. (2014) extend the definition of Vidović et al. (2011) by introducing time windows in which trucks have to reach locations. Popović et al. (2014) propose a variable neighborhood search heuristic approach to compute solutions for the extended formulation. Lai et al. (2013) allow street-turns; the authors study a problem with two different container sizes and a heterogeneous truck fleet (trucks can carry one or two containers at a time). In the problem formulation of Lai et al. (2013), trucks and containers cannot separate; additionally, importers have to be served before exporters. The last two restrictions again lead to a route size of four locations at the most. Lai et al. (2013) introduce a mathematical model and a heuristic approach based on the savings algorithm (Clarke and Wright, 1964). Schönberger et al. (2013) investigate a problem in which customers may be confronted with an imbalance of empty containers. Schönberger et al. (2013) distinguish between well-defined and flexible tasks, but the transportation of an empty container is not connected with the transportation of a fully loaded container (delivery and pickup trips) as it is the case in the mICT. Schönberger et al. (2013) introduce a mixed-integer programming (MIP) formulation that extends a less-than-truckload pickup and delivery model minimizing the total travel distance. The computational studies of Schönberger et al. (2013) establish the high complexity of the problem. Often, the solver failed to identify a feasible solution. Zhang et al. (2015) propose an almost similar problem to the mICT, in which a homogeneous fleet of trucks has to transport 20-foot and 40-foot empty and fully loaded containers between customers, one terminal and one depot within a given time period. The transportation request types and the definition of the depot, serving as parking place for the trucks and a sufficiently large storage for empty containers, do not distinguish from the mICT. However, not only the number of terminals is increased in the mICT: In the formulation of Zhang et al. (2015) trucks have to stay with the containers for (un-)loading operations and, additionally, no time windows are to be considered at customer locations and the terminal. A further assumption of Zhang et al. (2015) is the full-twin assumption, which restricts trucks carrying two 20-foot containers to finish both container transportation requests before starting a new request. The objective is to minimize the total traveling time of the trucks. Zhang et al. (2015) formulate a MIP as sequence-dependent amTSP with social constraints and propose three different tree search algorithms that are able to optimally solve small instances containing five to ten transportation requests and two to five trucks. To also solve instances of realistic size, the authors improve the reactive tabu search algorithm of Zhang et al. (2009) and apply it to randomly created instances and finally use it to compute solutions of a real-world instance. Again, the paper of Zhang et al. (2015) shows the difficulty to compute optimal solutions for container hinterland transportation: Although the authors do not consider time windows, the fastest of the three tree search algorithms takes nearly one hour to compute the optimal solution for the instance consisting of ten transportation requests and five trucks, while the slowest of the three algorithms does not find any solution. In comparison, Zhang et al. (2015) state that a fleet of 30 trucks can handle at most 75 containers in one day (Wang and Regan, 2002; Srour et al., 2010). The instances that are used in the literature consist of 75 (Zhang et al., 2010), 100 (Jula et al., 2005) and up to 200 transportation requests (Imai et al., 2007; Caris and Janssens, 2009; Braekers et al., 2013). 500 transportation requests and 93 trucks are contained in the real-world instance presented by Zhang et al. (2015) that is similar to the one presented by Chung et al. (2007).

The model proposed in this paper combines an extension of our recent graph representation (Funke and Kopfer, 2016) with successful techniques for modeling hinterland transportation (Jula et al., 2005; Zhang et al., 2010; Nossack and Pesch, 2013) as well as an approach for considering precedences in the field of maintenance (Goel and Meisel, 2013). Although there are models for solving transportation problems of 20-foot and 40-foot containers in the hinterland presented in literature (Chung et al., 2007; Vidović et al., 2012; Lai et al., 2013), these models refrain from considering the possibility of decoupling containers. The complexity of the problem increases a lot by considering the option of decoupling containers at customers' locations and thus enabling truck routes to contain more than four transportation requests (Schönberger et al., 2013).

## 3. Problem definition

The mICT is an extension of the ICT (Zhang et al., 2010). The objective of the presented problem is to optimize either the total travel distance, or the total operating time of the homogeneous fleet of trucks of one trucking company. The principle task of the company is to transport cargo. However, as empty containers are considered as transportation media for cargo, a distinction is made between three different types of tasks the trucking company has to deal with: empty container transportation, container handling (i.e. (un-)loading a container) and cargo transportation. The different tasks can be combined to one multi-task unit, a *hinterland request*. Zhang et al. (2010) mainly differentiate between two types of hinterland requests, *outbound full (OF)* and *inbound full (IF)* hinterland requests, each consisting of three tasks. While an OF hinterland request is concerned with pre-haulage operations, an IF hinterland request involves tasks that are operated in the end-haulage.

OF hinterland request:

1. At the customer's location the need for one empty container of a specific size (20- or 40-foot) is specified. An empty container of correct size has to be moved to the customer's location from some unspecified place.
2. The empty container is loaded at the customer's location. It has to be decided if stay-with or drop-and-pick strategy will be applied to the loading operation; i.e., the question arises, whether truck and container should separate (a decoupling operation is performed) or not. If the drop-and-pick strategy is chosen, the container has to be coupled by either the same or another truck after the container has been filled.
3. The fully loaded container has to be carried from the customer's place to a specific terminal where it is decoupled from the truck.

IF hinterland request:

1. A fully loaded container staying at a specific terminal has to be coupled with a truck and then has to be carried to a given customer's location.
2. The fully loaded container is unloaded at the customer's location. It has to be decided if stay-with or drop-and-pick strategy will be used to perform the unloading operation. In case of drop-and-pick strategy, the container has to be coupled by either the same or another truck after the container has been emptied.
3. The empty container has to be carried from the customer's location to some unspecified location.

The first and third task concern movements of containers (empty or fully loaded) and therefore require the presence of a truck for carrying the container. The second task is a container handling operation. In this paper it is assumed that each customer (i.e., sender or receiver) has the opportunity to lift (i.e., to separate) a container from a truck. For each customer's location it has to be decided if stay-with or drop-and-pick procedure is applied to perform the corresponding (un-)loading operation. It is further assumed that the drop-and-pick procedure needs additional time to (de-)couple and separate truck and container. Hinterland requests of customers which are not able to lift containers from trucks can also be modeled by the request types which are presented below.

Due to imbalances in trade, some areas have a surplus of empty containers and others a shortage (Braekers et al., 2011). To react to this imbalance of empty containers in one region, Zhang et al. (2010) take two more types of hinterland requests into consideration. These requests comprise one empty container transportation task, specifying containers to leave (*outbound empty (OE)*) or enter (*inbound empty (IE)*) a region:

OE hinterland request:

1. An empty container has to be picked up somewhere and transported to the terminal.

IE hinterland request:

1. An empty container has to be picked up at the terminal and transported to some place.

There are given container pick-up and delivery times (hard time-windows) for customers and terminals. It is assumed that the second visit of a customer is not time restricted, if a drop-and-pick procedure is performed. The length of a truck's working shift is limited and for each hinterland request the size (20-foot or 40-foot) of the needed container is specified. Trucks have a capacity of two TEU and start and end their routes at one specific depot.

The mICT is a combination of two distinct optimization problems. On the one hand, a variation of the Pickup and Delivery Problem is included to construct routes for cargo transportation. Cargo transportation requests are well-defined tasks with a given point of origin and destination for full container transportation. On the contrary, empty container transportation requests are flexible tasks, either the point where to pick up or the point where to deliver an empty container is undetermined and, consequently, is a matter of optimization. Thus, on the other hand, empty containers have to be assigned to hinterland requests by solving an Assignment Problem. In this paper, empty containers are assumed to be arbitrarily exchangeable among each other. Besides a street-turn, i.e., the possibility of using containers for outbound hinterland requests directly after the inbound requests realized with these containers have been finished, empty containers can be stored in one single depot. It is assumed that the depot has sufficient capacity to store all empty containers for inbound hinterland requests and that it is capable of fulfilling all demands for empty containers for outbound hinterland requests.

## 4. Mathematical model

To model the mICT, it is separated into its constituent parts at first. In a second step, synchronization constraints between the different parts are introduced, thereby combining the constituents to one model that covers the complete problem.

### 4.1. Graph definition

A homogeneous fleet of trucks $\mathcal{T}$, a depot $d$ together with a time horizon $[0, H]$, in which trucks are allowed to leave and enter $d$, and an ordered set $\mathcal{R} := (1, 2, \ldots, |\mathcal{R}|)$ of hinterland requests are considered. $\mathcal{R}$ can be subdivided into the sets:

| | |
|---|---|
| $\mathrm{OF}_{40} \subseteq \mathcal{R}$ | OF 40-foot hinterland requests |
| $\mathrm{OF}_{20} \subseteq \mathcal{R}$ | OF 20-foot hinterland requests |
| $\mathrm{IF}_{40} \subseteq \mathcal{R}$ | IF 40-foot hinterland requests |
| $\mathrm{IF}_{20} \subseteq \mathcal{R}$ | IF 20-foot hinterland requests |
| $\mathrm{OE}_{40} \subseteq \mathcal{R}$ | OE 40-foot hinterland requests |
| $\mathrm{OE}_{20} \subseteq \mathcal{R}$ | OE 20-foot hinterland requests |
| $\mathrm{IE}_{40} \subseteq \mathcal{R}$ | IE 40-foot hinterland requests |
| $\mathrm{IE}_{20} \subseteq \mathcal{R}$ | IE 20-foot hinterland requests |
| $\mathrm{OF} \subseteq \mathcal{R}$ | OF hinterland requests |
| $\mathrm{IF} \subseteq \mathcal{R}$ | IF hinterland requests |
| $\mathrm{OE} \subseteq \mathcal{R}$ | OE hinterland requests |
| $\mathrm{IE} \subseteq \mathcal{R}$ | IE hinterland requests |

To simplify the indexing the requests in $\mathrm{IF}_{20} \cup \mathrm{OF}_{20}$ stand in front of the requests in $\mathcal{R}$. Each request $i \in \mathcal{R}$ is given two locations $s_i,\ e_i$ and two time windows $\left[a^{s_i}, b^{s_i}\right]$ and $\left[a^{e_i}, b^{e_i}\right]$, in which a truck has to visit the corresponding location for the first time. For a request $i \in \mathrm{OF/IF}$, $s_i$ is equal to the sender's/terminal's location and $e_i$ to the terminal's/receiver's location that are specified by $i$. For a request $i \in \mathrm{IE} \cup \mathrm{OE}$, $s_i$ and $e_i$ are both equal to the terminal specified by $i$. Additionally, a request $i \in \mathrm{IF} \cup \mathrm{OF}$ is given a time needed for (un-)loading a container; this time is denoted by $\mathrm{load}_i$. Times for (de-)coupling are assumed to be the same for each customer; these times are denoted by cpl/dcpl.

The models of the underlying sub-problems of the mICT can be defined on a weighted directed graph $G = (V, A)$ where $V$ represents the node set and $A$ the arc set. A node $v \in V$ is a representative of a task together with one location where the task starts and another location where the task ends. These two locations do not have to be different. A task, which is represented by a node, is either a task taking place at the depot or a subtask of a hinterland request, such a task might contain a (de-)coupling operation. A node $v \in V$ is given a location pair $(\mathrm{orig}_v, \mathrm{dest}_v)$, where the represented task starts and ends, a time window $[\mathrm{start}_v, \mathrm{end}_v]$, in which a truck has to reach the node, and a service time $(\mathrm{serv}_v)$ that a truck requires to perform the task, i.e., the time a truck has to stay at the node. An arc $(v, w) \in A$ represents the tasks that take place between $v$ and $w$. The majority of arcs represents the movement from the ending location $\mathrm{dest}_v$ of the arc's tail to the starting location $\mathrm{orig}_w$ of the arc's head together with a (de-)coupling operation that might be executed between $v$ and $w$, but an arc can also represent an (un-)loading operation. In the following figures, arcs representing moving operations are depicted by consistent lines and arcs representing (un-)loading operations are depicted by dashed lines.

Fig. 1 shows representations of different types of hinterland requests $i \in \mathcal{R}$. Operations of empty hinterland requests are represented by one single node. This node represents a decouple (OE) or couple (IE) operation taking place at the request's terminal and has to start within the terminal's time window. As in this paper it is assumed that truck and container can separate at every location of a customer, the representation for full container transportation is more complex; hinterland requests defining the movement of 20-foot and 40-foot containers are treated differently. Two nodes are necessary to represent the operations needed by a 40-foot hinterland request. We follow the idea of Jula et al. (2005) and introduce one

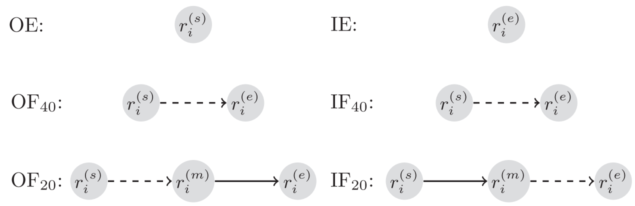


**Fig. 1.** Graph representation of the request types.

single node $r_i^{(e)}$ or $r_i^{(s)}$ for $i \in \mathrm{OF}_{40}$ or for $i \in \mathrm{IF}_{40}$ representing the transportation of a fully loaded 40-foot container from the customer's location to the terminal, or vice versa. Similar to the representation of an empty hinterland request, the node $r_i^{(s)}$ or $r_i^{(e)}$ for $i \in \mathrm{OF}_{40}$ or for $i \in \mathrm{IF}_{40}$ specifies the demand or supply for empty containers at the terminal. The main difference between 40-foot and 20-foot hinterland requests is that fully loaded 20-foot containers do not have to be transported on the direct route from customers to terminals, or vice versa. Thus, the origin–destination pairs of nodes representing fully loaded container transportation are disconnected in the case of 20-foot containers. Three nodes are obtained representing the entire tasks of $\mathrm{OF}_{20}$ and $\mathrm{IF}_{20}$ hinterland requests. One node states tasks taking place at the terminal, two nodes represent tasks taking place at the customer's location: One node includes tasks needed for empty containers and the other one includes tasks needed for fully loaded containers. The customer's location is duplicated as it might be visited two times by trucks if drop-and-pick procedure is used. Time windows at nodes representing $\mathrm{OF}_{20}$ and $\mathrm{IF}_{20}$ hinterland requests are the terminals' and customers' time windows; services times are equal to the time needed to (de-)couple containers. Time windows at nodes representing $\mathrm{OF}_{40}$ and $\mathrm{IF}_{40}$ hinterland requests can be obtained by shifting and intersecting time windows of customer and terminal (see, e.g. Zhang et al., 2010; Nossack and Pesch, 2013 for a detailed description of the representation of time windows); the service time needs to be extended by the sum of driving and (un-)loading durations.

Fig. 2 shows three different possibilities for (de-)coupling 20-foot containers from an inbound empty hinterland request $i$ and an outbound empty hinterland request $o$. The first possibility is depicted by straight arcs: A truck fulfills a street-turn by collecting an inbound empty container from $i$ and taking it to $o$. Then, the truck stays with the container while it is loaded (arc $(r_o^{(s)}, r_o^{(m)})$) and fetches the fully loaded container from the customer's location to the terminal (arc $(r_o^{(m)}, r_o^{(e)})$). A further possibility, in which a container is loaded via drop-and-pick procedure is depicted by the dotted arc. Here, a truck decouples a container at $o$'s sender's location (node $r_o^{(s)}$) for carrying out another task (like the collection of an empty container at the terminal of $i$) while the container is loaded. The possibility that is shown by zigzag arcs only holds for 20-foot containers: A truck does not move directly from the sender's location to the terminal of $o$ (arc $(r_o^{(m)}, r_o^{(e)})$), instead it chooses a detour by collecting the empty container of $i$ first and then moving with two containers to the terminal of $o$.

Most locations specify whether a (de-)coupling operation is to be fulfilled or not. Thus, the corresponding (de-)coupling operation is represented by the corresponding node. The only exceptions are decoupling operations that might accrue to fulfill an (un-)loading operation via drop-and-pick strategy what is to decide. If the drop-and-pick strategy is used, additional decoupling and coupling operations have to be performed. In our model, decoupling operations that might be needed for drop-and-pick strategy are included at arcs, and coupling operations that might be needed for drop-and-pick strategy are included at nodes. The set $A_L$ of dashed arcs in Fig. 1 represents (un-)loading operations that accrue between nodes:

$$A_L := \left\{ \left(r_i^{(s)}, r_i^{(e)}\right) | i \in \mathrm{OF}_{40} \cup \mathrm{IF}_{40} \right\} \cup \left\{ \left(r_i^{(s)}, r_i^{(m)}\right) | i \in \mathrm{OF}_{20} \right\} \cup \left\{ \left(r_i^{(m)}, r_i^{(e)}\right) | \ i \in \mathrm{IF}_{20} \right\}$$

A container is (un-)loaded via stay-with procedure if a truck traverses a loading arc in $A_L$. Otherwise, the container is (un-)loaded via drop-and-pick procedure. This means, that all arcs leaving the tail of an arc in $A_L$ and not entering its head, have to store a decoupling operation in addition to their included moving operation.

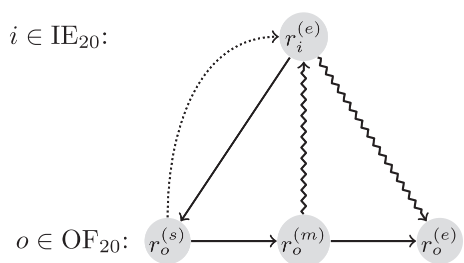


**Fig. 2.** Coupling possibilities.

To model the intermediate stops at a depot in a truck's route needed for container storage operations, we introduce $|\mathcal{R}| + 2|\mathcal{T}|$ nodes representing the depot. For each request $i \in \mathrm{OF}_{40} \cup \mathrm{OE}_{40}$ a depot node $d_i^{+40}/i \in \mathrm{OF}_{20} \cup \mathrm{OE}_{20}$ a depot node $d_i^{+20}/i \in \mathrm{IF}_{40} \cup \mathrm{IE}_{40}$ a depot node $d_i^{-40}/i \in \mathrm{IF}_{20} \cup \mathrm{IE}_{20}$ a depot node $d_i^{-20}$ is introduced. Duplicating the depot is based on an idea of Nossack and Pesch (2013) and enables a truck to visit the depot several times during its tour. The idea also states the assumption that the depot can provide an empty container for each outbound hinterland request and has sufficient capacity to store all empty containers obtained by inbound hinterland requests. For each truck $t \in \mathcal{T}$ two additional nodes $d_t^+$ and $d_t^-$ serve as points of origin and destination of $t$'s route. Depot duplicates have a time window equal to the time horizon $[0, H]$. An exemption applies to depots $d_i^{-40}, d_i^{-20}$ $(i \in \mathcal{R})$ which are needed for storage operations. These depots are allowed to start dcpl times before the time horizon. This exemption is made as there are more depot storage nodes introduced than normally needed by a solution to a problem instance. Thus, container coupling and decoupling operations that only take place in the depot should take place at zero duration time. Table 1 summarizes the considerations for nodes representing tasks taking place at the depot. Table 2 shows the values of the arcs. As coupling and decoupling operations are included on nodes, the duration of arcs $(d_i^{+40}, d_j^{-40})$ and $(d_i^{+20}, d_j^{-20})$ only set the time between storage operations taking place in the depot to zero.

For reasons of capacity, some arcs can be deleted from the graph as they are not contained in any feasible solution: Each truck that has coupled a 40-foot container has to decouple the container before it couples another container. Thus, nodes including a coupling operation of an empty 40-foot container can only be connected with nodes including a decoupling operation of an empty 40-foot container and vice versa. Reverse arcs $(w, v)$ of loading arcs $(v, w) \in A_L$ can be deleted, too.

An example of a graph for an instance consisting of one truck (i.e. $\mathcal{T} = \{t\}$) and three hinterland requests $\mathcal{R} = (1, 2, 3)$, with $\{1, 2\} \subseteq \mathrm{OE}_{40}$ and $\{3\} \subseteq \mathrm{IF}_{40}$ is depicted in Fig. 3. Truck $t$ has to start its route at $d_t^+$ and end at $d_t^-$. The node $r_3^{(s)}$ represents the transportation of a fully loaded container from the terminal to the receiver. Empty containers can be obtained from depot nodes $d_1^{+40}$ and $d_2^{+40}$ or the receiver's node $r_3^{(e)}$. Empty containers can be delivered to terminal nodes $r_1^{(s)}$ and $r_2^{(s)}$ or the depot node $d_3^{-40}$.

### 4.2. Assigning containers

In this section, the problem of assigning containers to requests is modeled as a Multicommodity Flow Problem (MFP). Three different sets of integral decision variables $x_{vw}^{(1)}, x_{vw}^{(2)}$ and $y_{vwc}$ corresponding to the containers' sizes and filling level are defined. The first two sets $x_{vw}^{(1)}/x_{vw}^{(2)}$ describe movements of empty 20-foot/40-foot containers on arc $(v, w) \in A$. Besides the containers' sizes, no distinction between empty containers can be drawn, as it is assumed that empty containers are arbitrarily interchangeable with each other. Decision variables indicating the movement of empty containers are defined as follows, whereby $\mathrm{teu}_c$ denotes the size in TEU of container $c$:

$$\forall k \in \{1, 2\} : x_{vw}^{(k)} := |\{c | c \text{ is an empty container with } \underset{c}{\mathrm{teu}} = k, \ \mathrm{traversing}(v, w)\}|$$

Because of the specified point of origin and destination of transported cargo, each fully loaded container can be distinguished. Due to Section 4.1, detours are only permitted for the transportation of fully loaded 20-foot containers. Thus, for each request $i \in \mathrm{IF}_{20} \cup \mathrm{OF}_{20}$ a specific container $c_i$ is introduced in the set $\mathcal{C}$ containing fully loaded 20-foot containers, i.e., $\mathcal{C} := \{c_i \,|\, i \in \mathrm{IF}_{20} \cup \mathrm{OF}_{20}\}$. The decision variables indicating the movement of fully loaded 20-foot containers are binary:

$$y_{vwc} := \begin{cases} 1, & \text{if arc } (v, w) \text{ is traversed by container } c \in \mathcal{C}, \\ 0, & \text{otherwise.} \end{cases}$$

When a specific request $i \in \mathcal{R}^F$ is considered, we also write $y_{wvc_i}$ with $c_i \in \mathcal{C}$ is the fully loaded 20-foot container corresponding to $i$.

For each node a multi-dimensional balance vector $b$ defining the demand/supply for empty 40-foot containers and empty/fully loaded 20-foot containers is introduced. $2 + |\mathrm{IF}_{20} \cup \mathrm{OF}_{20}|$ dimensions are needed: The first two dimensions define the demand/supply for empty containers and the remaining dimensions for fully loaded 20-foot containers, i.e.:

$$\forall v \in V, \ \forall k \in \{1, 2\} : b_v^k := \begin{cases} 1, & v \text{ provides a container } c \text{ with } \mathrm{teu}_c = k, \\ -1, & v \text{ needs a container } c \text{ with } \mathrm{teu}_c = k, \\ 0, & \text{otherwise.} \end{cases}$$

$$\forall v \in V, \ \forall k \in \{3, \dots, 2 + |\mathrm{IF}_{20} \cup \mathrm{OF}_{20}|\} : b_v^k := \begin{cases} 1, & v \text{ provides } c_{k-2} \in \mathcal{C}, \\ -1, & v \text{ needs } c_{k-2} \in \mathcal{C}, \\ 0, & \text{otherwise.} \end{cases}$$

A complete list of balance values is presented in Table 3. Not mentioned balance values are set to zero.

An arc is traversed by at most one truck that is able to carry at most two TEU. Thus, the container capacity $u_{vw}$ of an arc $(v, w) \in A$ is less than or equal to two. Although – in the case of drop-and-pick strategy – no truck has to traverse a loading arc $e_i \in A_L$, in any case the corresponding container $c_i$ that is (un-)loaded on arc $e_i$ traverses $e_i$. Thus, containers $c_i$ traversing

**Table 1**
Depot nodes.

| Node | Duration (serv) | Balance ($b$) | Time window ($[\text{start}_v, \text{end}_v]$) |
|---|---|---|---|
| $\{d_t^+ \mid t \in \mathcal{T}\}$ | 0 | $b \equiv 0$ | $[0, H]$ |
| $\{d_t^- \mid t \in \mathcal{T}\}$ | 0 | $b \equiv 0$ | $[0, H]$ |
| $d_i^{+40}$ | cpl | $b_2 = 1$ | $[0, H]$ |
| $d_i^{+20}$ | cpl | $b_1 = 1$ | $[0, H]$ |
| $d_i^{-40}$ | dcpl | $b_2 = -1$ | $[0 - \text{dcpl}, H]$ |
| $d_i^{-20}$ | dcpl | $b_1 = -1$ | $[0 - \text{dcpl}, H]$ |

**Table 2**
Arcs.

| Arc | Duration (dur) | Cost ($c$) |
|---|---|---|
| $i \in \text{OF}_{40},\ j \in \text{IF}_{40}$: | | |
| $(d_i^{+40}, d_j^{-40})$ | $-\text{cpl} - \text{dcpl}$ | 0 |
| $i \in \text{OF}_{20},\ j \in \text{IF}_{20}$: | | |
| $(d_i^{+20}, d_j^{-20})$ | $-\text{cpl} - \text{dcpl}$ | 0 |
| Loading arcs: | | |
| $e_i = (v, w) \in A_L$ | $\text{load}_i$ | 0 |
| $(v, w'),\ w' \neq w$ | $\text{dcpl} + \text{dist}(\text{dest}_v, \text{orig}_{w'})$ | $\text{dist}(\text{dest}_v, \text{orig}_{w'})$ |
| Remaining arcs | | |
| $(v, w)$ | $\text{dist}(\text{dest}_v, \text{orig}_w)$ | $\text{dist}(\text{dest}_v, \text{orig}_w)$ |

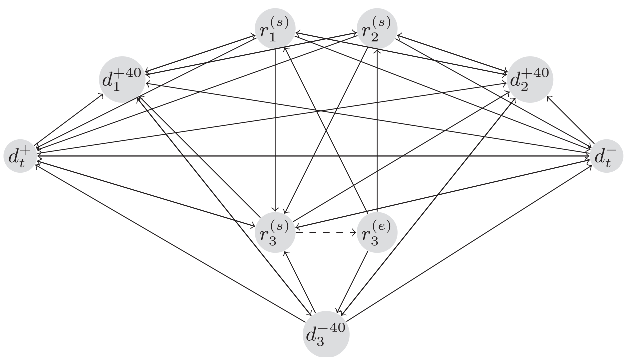


**Fig. 3.** The instance graph.

arc $e_i$ can be modeled implicitly by reducing $e_i$'s capacity by the container's size $\text{teu}_{c_i}$. Analogously, only empty trucks are allowed to enter a node representing the transportation of a fully loaded 40-foot container. It holds:

$$u_{vw} := \begin{cases} 2 - \text{teu}_{c_i}, & (v, w) = e_i \in A_L \\ 0, & w = r_i^{(s)},\ i \in \text{IF}_{40} \text{ or } w = r_i^{(e)},\ i \in \text{OF}_{40} \\ 2, & \text{otherwise} \end{cases}$$

Fig. 4 shows balance values of nodes and capacities of arcs for an instance consisting of three requests $\mathcal{R} = (r_1, r_2, r_3)$, whereby $r_1/r_2$ represent inbound/outbound full 20-foot container hinterland requests and $r_3$ is an outbound full 40-foot container hinterland request. Again, dashed arcs represent loading arcs. For dashed arcs, the original capacity of two is decremented by the size of the container that is (un-)loaded on the arc, because in any case the container traverses the arc for (un-)loading. If a truck traverses an arc where a 40-foot respectively 20-foot container is (un-)loaded, then reducing the capacity of the arc ensures that this truck has not loaded any further container respectively has loaded at most one further 20-foot container. Each full 20-foot container hinterland request induces an additional dimension to the balance vector, enabling that the assigned fully loaded container is transported between its customer and terminal or vice versa. An additional depot for storage operations is introduced for each hinterland request, satisfying the assumption that the depot has enough storage capacity for inbound hinterland requests and can provide a container for each outbound hinterland request. The additional depots can be visited but need not to be visited. For example, if a truck fulfills a street-turn between $r_1$ and $r_2$ by moving between $r_1^{(e)}$ and $r_2^{(s)}$, then $d_2^{+20}$ and $d_1^{-20}$ are also connected without inducing any extra costs or time. However, the

**Table 3**
Request nodes.

| Node $v$ | Locations ($\mathrm{orig}_v, dest_v$) | Service time ($\mathrm{serv}_v$) | Balance ($b_v$) | Time window ($[\mathrm{start}_v, \mathrm{end}_v]$) |
|---|---|---|---|---|
| $r_i \in \mathrm{OF}_{40}$: | | | | |
| $r_i^{(s)}$ | $\mathrm{orig} = \mathrm{dest} = s_i$ | 0 | $b_2 = -1$ | $[a^{s_i}, b^{s_i}]$ |
| $r_i^{(e)}$ | $\mathrm{orig} = s_i,\ \mathrm{dest} = e_i$ | $\mathrm{cpl} + \mathrm{dist}(s_i, e_i) + \mathrm{dcpl}$ | $b \equiv 0$ | $[a^{e_i} - \mathrm{dist}(s_i, e_i), b^{e_i} - \mathrm{dist}(s_i, e_i)]$ |
| $r_i \in \mathrm{IF}_{40}$: | | | | |
| $r_i^{(s)}$ | $\mathrm{orig} = s_i,\ \mathrm{dest} = e_i$ | $\max\{a^{e_i} - b^{s_i}, \mathrm{cpl} + \mathrm{dist}(s_i, e_i)\}$ | $b \equiv 0$ | $[\min\{\max\{a^{s_i}, a^{e_i} - \mathrm{cpl} - \mathrm{dist}(s_i, e_i)\}, b^{s_i}\}, \min\{b^{s_i}, b^{e_i} - \mathrm{cpl} - \mathrm{dist}(s_i, e_i)\}]$ |
| $r_i^{(e)}$ | $\mathrm{orig} = \mathrm{dest} = e_i$ | cpl | $b_2 = 1$ | $[0, H]$ |
| $r_i \in \mathrm{OF}_{20}$: | | | | |
| $r_i^{(s)}$ | $\mathrm{orig} = \mathrm{dest} = s_i$ | 0 | $b_1 = -1$ | $[a^{s_i}, b^{s_i}]$ |
| $r_i^{(m)}$ | $\mathrm{orig} = \mathrm{dest} = s_i$ | cpl | $b_{i+2} = 1$ | $[0, H]$ |
| $r_i^{(e)}$ | $\mathrm{orig} = \mathrm{dest} = e_i$ | dcpl | $b_{i+2} = -1$ | $[a^{e_i}, b^{e_i}]$ |
| $r_i \in \mathrm{IF}_{20}$: | | | | |
| $r_i^{(s)}$ | $\mathrm{orig} = \mathrm{dest} = s_i$ | cpl | $b_{i+2} = 1$ | $[a^{s_i}, b^{s_i}]$ |
| $r_i^{(m)}$ | $\mathrm{orig} = \mathrm{dest} = e_i$ | 0 | $b_{i+2} = -1$ | $[a^{e_i}, b^{e_i}]$ |
| $r_i^{(e)}$ | $\mathrm{orig} = \mathrm{dest} = e_i$ | cpl | $b_1 = 1$ | $[0, H]$ |
| $r_i \in \mathrm{OE}_{40}$: | | | | |
| $r_i^{(s)}$ | $\mathrm{orig} = \mathrm{dest} = s_i$ | dcpl | $b_2 = -1$ | $[a^{s_i}, b^{s_i}]$ |
| $r_i \in \mathrm{OE}_{20}$: | | | | |
| $r_i^{(s)}$ | $\mathrm{orig} = \mathrm{dest} = s_i$ | dcpl | $b_1 = -1$ | $[a^{s_i}, b^{s_i}]$ |
| $r_i \in \mathrm{IE}_{40}$: | | | | |
| $r_i^{(e)}$ | $\mathrm{orig} = \mathrm{dest} = s_i$ | cpl | $b_2 = 1$ | $[a^{s_i}, b^{s_i}]$ |
| $r_i \in \mathrm{IE}_{20}$: | | | | |
| $r_i^{(e)}$ | $\mathrm{orig} = \mathrm{dest} = s_i$ | cpl | $b_1 = 1$ | $[a^{s_i}, b^{s_i}]$ |

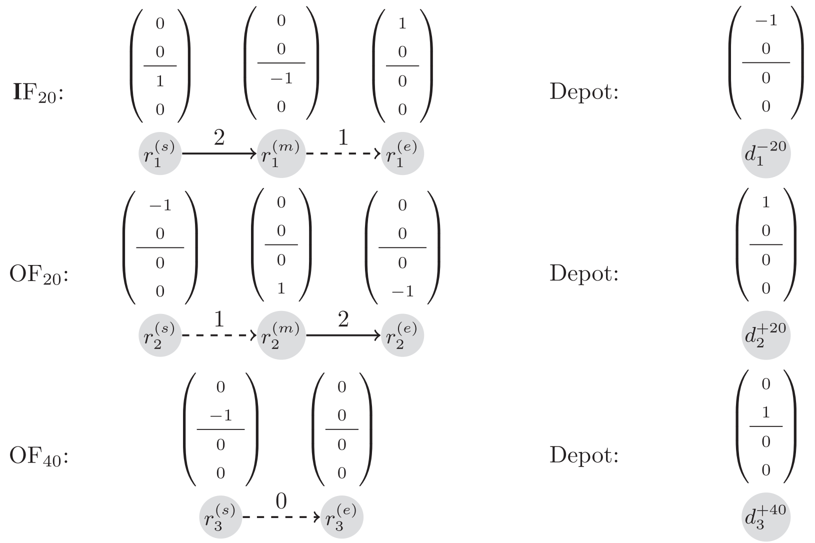


**Fig. 4.** Balances and capacities.

container storage operations of $r_1$ and $r_2$ can also take place at the depot. In any case, at the beginning of an outbound hinterland request an empty container is needed and at the end of an inbound hinterland request an empty container is obtained. Because we assume that empty 20-foot and 40-foot containers are interchangeable with each other, the first two dimensions of a balance vector only determine the size of the empty container that is needed or obtained at a node. Compared to the transportation of fully loaded containers only two dimensions representing the different sizes (20-foot and 40-foot) are needed to model the transportation of empty containers, instead of introducing an additional dimension for every container.

The problem of assigning containers to requests can be stated as follows.

$$\sum_{(v,w)\in A} x_{vw}^{(k)} - \sum_{(w,v)\in A} x_{wv}^{(k)} = b_v^{(k)} \quad \forall v \in V,\ \forall k \in \{1,2\} \tag{1}$$

$$\sum_{(v,w)\in A} y_{vwc_i} - \sum_{(w,v)\in A} y_{wvc_i} = b_v^{(i+2)} \quad \forall v \in V,\ \forall i \in \mathrm{IF}_{20} \cup \mathrm{OF}_{20} \tag{2}$$

$$\sum_{k\in\{1,2\}} k \cdot x_{vw}^{(k)} + \sum_{c\in\mathcal{C}} y_{vwc} \leqslant u_{vw} \quad \forall (v,w) \in A \tag{3}$$

$$x_{vw}^{(k)} \in \{0,1,2\} \quad \forall (v,w) \in A,\ \forall k \in \{1,2\} \tag{4}$$

$$y_{vwc} \in \{0,1\} \quad \forall (v,w) \in A,\ \forall c \in \mathcal{C} \tag{5}$$

Constraints (1) define the supply/demand for empty containers and Constraints (2) define the supply/demand for fully loaded 20-foot containers. Constraints (3) ensure that each arc is traversed by at most two TEU. The domains of the decision variables are given by Constraints (4) and (5).

### 4.3. Building routes

The sub-problem of building truck routes is presented as an extension of an amTSPTW formulation. As mentioned at the beginning of Section 4.1, each node is visited exactly once. Locations that might be visited more than once are duplicated in the node set. The only exceptions are the depot duplicates $d_t^+, d_t^-$ $(t \in \mathcal{T})$ which represent the points of origin and destination of a truck's route. $d_t^+$ is left exactly once and $d_t^-$ is entered exactly once. For each node $v \in V,\ t_v$ states the point in time the location represented by $v$ is reached by a truck; the arc variable $\delta_{vw}$ determines whether a truck traverses an arc or not:

$$\delta_{vw} := \begin{cases} 1, & \text{if arc } (v,w) \text{ is traversed by a truck} \\ 0, & \text{otherwise} \end{cases}$$

The problem of constructing routes can be stated as follows.

$$\sum_{(v,w)\in A}\delta_{vw}=\sum_{(w,v)\in A}\delta_{wv}=1\quad\forall v\in V\setminus\{d_t^+,d_t^-\,|\,t\in\mathcal{T}\}\tag{6}$$

$$\sum_{(v,w)\in A}\delta_{vw}=1,\quad\sum_{(w,v)\in A}\delta_{wv}=0\quad\forall v\in\{d_t^+\,|\,t\in\mathcal{T}\}\tag{7}$$

$$\sum_{(v,w)\in A}\delta_{vw}=0,\quad\sum_{(w,v)\in A}\delta_{wv}=1\quad\forall v\in\{d_t^-\,|\,t\in\mathcal{T}\}\tag{8}$$

$$\delta_{vw}=1\Rightarrow t_v+\mathrm{dur}_{vw}+\mathrm{serv}_v\leqslant t_w\quad\forall(v,w)\in A\setminus A_L\tag{9}$$

$$\mathrm{start}_v\leqslant t_v\leqslant\mathrm{end}_v\quad\forall v\in V\setminus\left\{r_i^{(e)}\,|\,i\in\mathrm{OF}_{40}\right\}\tag{10}$$

$$t_v\in\mathbb{R}\quad\forall v\in V\tag{11}$$

$$\delta_{vw}\in\{0,1\}\quad\forall(v,w)\in A\tag{12}$$

Constraints (6) ensure that nodes are visited exactly once. Constraints (7) and (8) ensure that starting/ending depot nodes are left/entered exactly once. Subtours are eliminated and time windows are set by Constraints (9) and (10). Finally, Constraints (11) and (12) set the domains of the decision variables.

### 4.4. Coupling of the models

To combine the two formulations of the former sections it is to ensure on the one hand that containers are moved by trucks and on the other hand that times needed for (un-)loading operations are observed.

$$\sum_{k\in\{1,2\}}\underset{c}{\mathrm{teu}}\cdot x_{vw}^{(k)}+\sum_{c\in\mathcal{C}}y_{vwc}\leqslant 2\cdot\delta_{vw}\qquad\forall(v,w)\in A\tag{13}$$

$$t_v+(1-\delta_{vw})\cdot\mathrm{dcpl}+\underset{vw}{\mathrm{dur}}+\underset{v}{\mathrm{serv}}-\delta_{vw}\cdot\mathrm{cpl}\leqslant t_w\qquad\forall(v,w)\in A_L\tag{14}$$

$$\underset{r_i^{(e)}}{\mathrm{start}}-\left(1-\delta_{r_i^{(s)}r_i^{(e)}}\right)\cdot\mathrm{cpl}\leqslant t_{r_i^{(e)}}\qquad\forall i\in\mathrm{OF}_{40}\tag{15}$$

$$t_{r_i^{(e)}}\leqslant\underset{r_i^{(e)}}{\mathrm{end}}-\left(1-\delta_{r_i^{(s)}r_i^{(e)}}\right)\cdot\mathrm{cpl}\qquad\forall i\in\mathrm{OF}_{40}\tag{16}$$

Constraints (13) ensure that each container flow is covered by a truck. The implementation of constraints representing (un-)loading operations of containers is inspired by Goel and Meisel (2013) who show a mTSPTW formulation of a maintenance problem of electricity networks including precedence constraints. Constraints (14) implement the two possibilities for (un-)loading a container. If a loading arc $(v,w)\in A_L$ of a full hinterland request $i\in\mathcal{R}$ is not traversed by any truck ($\delta_{vw}=0$), then the fully loaded container specified by $i$ is (un-)loaded by drop-and-pick procedure. This means that between the starting times of $v$ and $w$ some precedence time has to take place to ensure that the former task (i.e., decoupling and (un-)loading the container) is completed before the latter (i.e., collecting the container) starts. The stay-with procedure applies to (un-)loading operations, when a truck traverses arc $(v,w)\in A_L$, i.e. $\delta_{vw}=1$. In this case the precedence time has to be decreased by the time that is needed to couple a container as the service time of node $w$ already includes this additional coupling duration. Besides the nodes $r_i^{(e)},i\in\mathrm{OF}_{40}$, this additional decreasing has no impact on the nodes' time windows as it is assumed that a truck is allowed to visit a customer's location for the second time at any point in time within the time horizon. But nodes $r_i^{(e)},i\in\mathrm{OF}_{40}$ additionally represent the transportation of a fully loaded container from the sender's location to the terminal, thus, the arrival time of a truck at these nodes depends not only the sender's time window but also on the terminal's time window. Constraints (15) and (16) set the different time windows of nodes $r_i^{(e)},i\in\mathrm{OF}_{40}$, in case the stay-with or drop-and-pick procedure is used.

Fig. 4 shows that the capacity of a loading arc is decremented by the size of the container that is (un-)loaded on the arc. Meaning, that (un-)loading a container is only implicitly modeled by decreasing the arc's capacity. That is reason why container flows on loading arcs $A_L$ are not excluded from Constraints (13). A truck can traverse a loading arc independently from the fact that this arc is traversed in either case by the container that is (un-)loaded at this arc. As the distance of an loading arc is always zero (the point of origin and destination represent the location of the same customer), it is not improbable that a truck traverses a loading arc. Decreasing the arcs' capacity also ensures a truck transporting at most one further 20-foot container/no further container, if it stays with a 20-foot/40-foot container for (un-)loading.

### 4.5. The objective

Two different objectives are considered. The objective minimizing the total distance that is traveled by trucks can be stated as follows, whereby the last two summands are constants representing the carriage of fully loaded 40-foot containers:

$$\min\sum_{e\in A}c_e\cdot\delta_e+\sum_{i\in\mathrm{OF}_{40}}\mathrm{dist}\left(\underset{r_i^{(e)}}{\mathrm{orig}},\underset{r_i^{(e)}}{\mathrm{dest}}\right)+\sum_{i\in\mathrm{IF}_{40}}\mathrm{dist}\left(\underset{r_i^{(s)}}{\mathrm{orig}},\underset{r_i^{(s)}}{\mathrm{dest}}\right)\tag{17}$$

The objective minimizing the total service time of the trucks can be stated by comparing the time trucks leave starting depots with the time trucks enter ending depots:

$$\min \sum_{v \in \left\{ d_t^- \mid t \in \mathcal{T} \right\}} t_v - \sum_{v \in \left\{ d_t^+ \mid t \in \mathcal{T} \right\}} t_v \tag{18}$$

## 5. Computational studies

The presented model is implemented in C++ using IBM ILOG CPLEX Studio version 12.5.1. Tests have been run on an Intel Core i5-3230, 2.6 GHz machine. The time to solve one single instance is limited to one hour.

### 5.1. Instances with up to six hinterland requests

A first test is made that shows the complexity of the considered problem. Instances consisting of at most six hinterland requests are solved by an implementation of the model. The two objectives minimizing operating time and travel distance are compared.

#### 5.1.1. Test instances

Randomly created test instances which differ in the number of OF, IF, OE and IE hinterland requests are used:

- $|\mathrm{OF}| + |\mathrm{IF}| = 0,\ |\mathrm{OE}| + |\mathrm{IE}| = 6$ (cases_0_6)
- $|\mathrm{OF}| + |\mathrm{IF}| = 1,\ |\mathrm{OE}| + |\mathrm{IE}| = 2$ (cases_1_2)
- $|\mathrm{OF}| + |\mathrm{IF}| = 1,\ |\mathrm{OE}| + |\mathrm{IE}| = 3$ (cases_1_3)
- $|\mathrm{OF}| + |\mathrm{IF}| = 2,\ |\mathrm{OE}| + |\mathrm{IE}| = 2$ (cases_2_2)

The different locations are randomly placed in the plane $[0, 10] \times [0, 10]$. Distances between locations are computed by measuring the euclidean distances between the locations, then multiplying them by 1000 and obtaining integer numbers by cutting the results before the first decimal. The longest distance between two different locations is about four hours. The triangle inequality holds for each instance. The duration of (un-)loading a container is a random number varying between 20 min and two hours.

While request types, (un-)loading durations and locations do not change within a test-set, time windows (none, three different instance types with realistic, three different instance types with randomly chosen) and numbers of trucks (1,2,3) vary, so that 84 different test instances are obtained. The realistic time windows are defined according to Van Der Horst and De Langen (2008), who state that senders and receivers have limited opening hours with peak times between 6.00–9.00 a.m. and 5.00–8.00 p.m. The time horizon is set from 0 a.m. to 24 p.m.

We obtain two different test-sets by setting the time for coupling and decoupling a container to 5 min and afterward to 20 min.

#### 5.1.2. Results

The results for the different objectives "travel distance" and "operating time" are listed in Tables 4 and 5. In the rows, the summarized and rounded results for all instance sets (first row) and the results differentiated for each instance type (following rows) are listed. The first three columns show absolute values representing the number of instances which have been solved to optimality ("Opt"), solved with an optimality gap after one hour ("Feas") and unsolved instances ("Inf"). In Table 5, the next two columns present averaged values for those instances for which a solution could be obtained. The column "Serv" gives the operating time and "Km" the traveled kilometers. As the objective minimizing the total travel distance has no impact on the operating time, the column "Serv" is excluded in Table 4. The last but one column gives an average value

**Table 4**
Results "travel distance", summary.

| Instance | Opt | Feas | Inf | Km | Gap | Time |
|---|---|---|---|---|---|---|
| 5 min, all | 80 | 0 | 4 | 21,716 | 0 | 16 |
| 5 min, cases_0_6 | 19 | 0 | 2 | 28,714 | 0 | 6 |
| 5 min, cases_1_2 | 21 | 0 | 0 | 8484 | 0 | 2 |
| 5 min, cases_1_3 | 21 | 0 | 0 | 24,104 | 0 | 2 |
| 5 min, cases_2_2 | 19 | 0 | 2 | 26,705 | 0 | 5 |
| 20 min, all | 76 | 0 | 8 | 21,293 | 0 | 15 |
| 20 min, cases_0_6 | 18 | 0 | 3 | 28,423 | 0 | 4 |
| 20 min, cases_1_2 | 21 | 0 | 0 | 8484 | 0 | 2 |
| 20 min, cases_1_3 | 19 | 0 | 2 | 24,104 | 0 | 2 |
| 20 min, cases_2_2 | 18 | 0 | 3 | 26,139 | 0 | 6 |

**Table 5**
Results "operating time", summary.

| Instance | Opt | Feas | Inf | Serv | Km | Gap | Time |
|---|---|---|---|---|---|---|---|
| 5 min, all | 77 | 3 | 4 | 29,651 | 23,201 | 18.7 | 21,493 |
| 5 min, cases_0_6 | 16 | 3 | 2 | 38,414 | 30,053 | 78.6 | 16,547 |
| 5 min, cases_1_2 | 21 | 0 | 0 | 13,440 | 9292 | 0 | 23 |
| 5 min, cases_1_3 | 21 | 0 | 0 | 30,982 | 24,912 | 0 | 80 |
| 5 min, cases_2_2 | 19 | 0 | 2 | 37,332 | 29,829 | 0 | 4843 |
| 20 min, all | 73 | 3 | 8 | 34,812 | 22,982 | 16.0 | 19,397 |
| 20 min, cases_0_6 | 15 | 3 | 3 | 46,438 | 30,373 | 67.5 | 14,538 |
| 20 min, cases_1_2 | 21 | 0 | 0 | 16,954 | 9696 | 0 | 19 |
| 20 min, cases_1_3 | 19 | 0 | 2 | 36,432 | 25,591 | 0 | 78 |
| 20 min, cases_2_2 | 18 | 0 | 3 | 42,309 | 28,338 | 0 | 4762 |

**Table 6**
Results "travel distance", summary.

| Instance | Opt | Feas | Inf | Km | Gap | Time |
|---|---|---|---|---|---|---|
| All | 31 | 19 | 10 | 649 | 4.5 | 122,620 |
| DS1 | 5 | 1 | 0 | 497 | 0.8 | 3622 |
| DS2 | 0 | 2 | 4 | 701 | 4.7 | 24,763 |
| DS3 | 3 | 3 | 0 | 726 | 6.5 | 10,822 |
| DS4 | 2 | 3 | 1 | 717 | 7.9 | 25,909 |
| DS5 | 4 | 2 | 0 | 646 | 2.9 | 7275 |
| DS6 | 4 | 1 | 1 | 641 | 3.1 | 10,355 |
| DS7 | 4 | 1 | 1 | 667 | 2.7 | 8290 |
| DS8 | 2 | 3 | 1 | 617 | 11.0 | 11,881 |
| DS9 | 4 | 2 | 0 | 576 | 3.3 | 7642 |
| DS10 | 3 | 1 | 2 | 789 | 3.1 | 12,061 |

for the maximum gap to optimum as a percentage, excluding instances which could not be solved. The time in seconds that is needed to solve all instances of one row is stated in the last column.

There are four infeasible instances for the test cases in Table 4, in which the (de-)coupling times are set to 5 min. As the time horizon is not enlarged for the test-sets, in which the (de-)coupling duration is 20 min, the number of infeasible instances doubled in those test-sets. The objective value of the 20 min test-sets is better than the objective value of the 5 min test-sets, because the objective value is an average value, in which infeasible instances are excluded. The implementation that uses the objective minimizing the travel distance is able to optimally solve all of the other 80/76 instances. Three instances with six requests cannot be solved to optimality using the objective that minimizes the operating time. Altogether, minimizing the total travel distance seems to be much easier than minimizing the total operating time. While minimizing the travel distance of all 84 test instances takes about sixteen seconds, minimizing the total operating time needs between five and six hours.

### *5.2. Instances with eleven hinterland requests*

Due to the promising results for the running time of the implementation using the objective that minimizes the total distance of trucks, we perform tests on larger instances that are modifications of instances from literature sources. In these tests we only consider the implementation minimizing the total distance.

#### *5.2.1. Test instances*

We modify the five truck and eleven hinterland request instances of Sterzik and Kopfer (2013). Sterzik and Kopfer (2013) consider the ICT. To verify the quality of their proposed heuristic algorithm, the authors create ten instances by choosing randomly geographical data and time windows of customers, terminals and depots from Solomon's RC1-VRPTW-data sets (Solomon, 1987). The eleven hinterland requests consist of five OF and five IF hinterland requests and one IE hinterland request. As the ICT includes more than one depot, we modify the instances of Sterzik and Kopfer (2013) by considering only one depot that is located at its original position (Solomon, 1987). Sterzik and Kopfer (2013) round the euclidean distances to obtain integral values. In case that the triangle inequality does not hold, we modify distances to enforce that the triangle inequality is satisfied. We also take (de-)coupling durations that always have the value of two into account. Times for (un-)loading a container are set to ten. Thus, the time windows of terminals of OF and IF hinterland requests are not only shifted by the driving duration between customer and terminal, as it is the case for instances of Sterzik and Kopfer (2013), but also by the time that is needed to (un-)load a container. The additional service time leads to the fact that all but three instances are feasible. The number of trucks of the instance sets *DS3*, *DS7* and *DS10* is incremented by one so that a feasible solution also exists for these instances. We modify the number of $OF_{40}/OF_{20}$ and $IF_{40}/IF_{20}$ requests within the

**Table 7**
Results "travel distance", DS1, DS5, DS9.

| Instance | Opt | Feas | Inf | Km | Gap | Time |
|---|---|---|---|---|---|---|
| DS1 | 5 | 1 | 0 | 497 | 0.8 | 3622 |
| DS1.0_11 | 0 | 1 | 0 | 408 | 4.9 | 3599 |
| DS1.3_8 | 1 | 0 | 0 | 469 | 0 | 9 |
| DS1.5_6 | 1 | 0 | 0 | 545 | 0 | 1 |
| DS1.6_5 | 1 | 0 | 0 | 539 | 0 | 13 |
| DS1.8_3 | 1 | 0 | 0 | 525 | 0 | 0 |
| DS1.11_0 | 1 | 0 | 0 | 493 | 0 | 0 |
| DS5 | 4 | 2 | 0 | 646 | 2.9 | 7275 |
| DS5.0_11 | 0 | 1 | 0 | 573 | 12.4 | 3599 |
| DS5.3_8 | 0 | 1 | 0 | 688 | 4.9 | 3600 |
| DS5.5_6 | 1 | 0 | 0 | 760 | 0 | 51 |
| DS5.6_5 | 1 | 0 | 0 | 631 | 0 | 25 |
| DS5.8_3 | 1 | 0 | 0 | 651 | 0 | 0 |
| DS5.11_0 | 1 | 0 | 0 | 573 | 0 | 0 |
| DS9 | 4 | 2 | 0 | 576 | 3.3 | 7642 |
| DS9.0_11 | 0 | 1 | 0 | 541 | 14.1 | 3599 |
| DS9.3_8 | 0 | 1 | 0 | 594 | 5.4 | 3600 |
| DS9.5_6 | 1 | 0 | 0 | 600 | 0 | 146 |
| DS9.6_5 | 1 | 0 | 0 | 542 | 0 | 6 |
| DS9.8_3 | 1 | 0 | 0 | 575 | 0 | 0 |
| DS9.11_0 | 1 | 0 | 0 | 602 | 0 | 291 |

instances, while the geographical data, (un-)loading and (de-)coupling times stay the same. We obtain 60 instances (every case contains the ten instances of Sterzik and Kopfer (2013)):

- $|OF_{40}| + |IF_{40}| = 0,\ |OF_{20}| + |IF_{20}| + |IE_{20}| = 11$ (cases_0_11)
- $|OF_{40}| + |IF_{40}| = 3,\ |OF_{20}| + |IF_{20}| + |IE_{20}| = 8$ (cases_3_8)
- $|OF_{40}| + |IF_{40}| = 5,\ |OF_{20}| + |IF_{20}| + |IE_{20}| = 6$ (cases_5_6)
- $|OF_{40}| + |IF_{40}| + |IE_{40}| = 6,\ |OF_{20}| + |IF_{20}| = 5$ (cases_6_5)
- $|OF_{40}| + |IF_{40}| + |IE_{40}| = 8,\ |OF_{20}| + |IF_{20}| = 3$ (cases_8_3)
- $|OF_{40}| + |IF_{40}| + |IE_{40}| = 11,\ |OF_{20}| + |IF_{20}| = 0$ (cases_11_0)

#### 5.2.2. Results

Within one hour, the implementation minimizing the total travel distance is able to solve instances containing up to eleven hinterland requests. Compared to Sterzik and Kopfer (2013) the number of requests in the test instances is equal but adding the possibility of different sized containers to the problem definition increases the complexity a lot. Table 6 summarizes the results for all instances.

Table 7 shows the results for the instances *DS1*, *DS5* and *DS9* in more detail. As optimal solutions of instances in the set *cases_11_0* are always contained in the solution spaces of instances in the set *cases_0_11*, the objective value either decreases when considering 20-foot containers (*DS1*, *DS9*) or stays the same as for 40-foot containers (*DS5*). The additional degrees of freedom for instances containing more 20-foot than 40-foot containers usually lead to larger running times for those instances. The solution quality of cases in which 20-foot and 40-foot containers are considered together depends on the fact whether or not the size of containers differs for containers whose hinterland request pairs fit well together. The exclusion of combining possibilities is reason to the fact that the objective value decreases for most of the instances, in which different sized containers are considered.

## 6. Conclusion

This paper studies a problem arising in the pre- and end-haulage of an intermodal container transportation chain. Containers of different filling levels (fully loaded and empty) have to be transported between terminals, receivers, senders and one depot. While fully loaded containers are given points of origin and destination, the specification of a transportation request of an empty container is incomplete regarding either the point of origin or destination. An assignment problem as to which empty containers to use for which cargo request still has to be solved as well as the routing problem for trucks carrying the containers. The commonly considered issue of transporting 40-foot containers only is extended by introducing two different kinds of commodities, namely 40-foot and 20-foot containers. A mathematical model that simultaneously solves the combined problem of assigning containers to requests and building routes for trucks has been formulated and implemented. Small randomly created instances can be solved using the implementation with two different objectives (minimizing total distance and total operating time of trucks). Minimizing the total travel distance is much faster, so that the corresponding implementation is also applied to solve modified instances from literature sources that are about twice the size of the former instance set. To the best of our knowledge, in this paper a mathematical model and its implementation that are able to optimally solve problem instances of the studied problem type have been presented for the first time, even though these instances by nature are small.

Future research should focus on creating a heuristic approach to solve instances consisting of more than eleven hinterland transportation requests. For small-size instances the heuristic approach can be evaluated by using the obtained results of the presented model. An idea for a neighborhood search to instances containing no time windows is given by Funke et al. (2015). This heuristic approach considers three different neighborhoods, in which small container assignment problems are solved through a MIP, while an insertion method is used for building routes for trucks.

## References


Applegate, D., Bixby, R.E., Chvátal, V., Cook, W., 1995. Finding cuts in the TSP (A preliminary report), vol. 95-05. Citeseer.

Applegate, D., Bixby, R.E., Chvátal, V., Cook, W., 2011. The Traveling Salesman Problem: A Computational Study. Princeton University Press.

Bektas, T., 2006. The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega 34 (3), 209–219.

Braekers, K., Janssens, G.K., Caris, A., 2011. Challenges in managing empty container movements at multiple planning levels. Transp. Rev. 31 (6), 681–708.

Braekers, K., Caris, A., Janssens, G.K., 2013. Integrated planning of loaded and empty container movements. OR Spectrum 35 (2), 457–478.

Caris, A., Janssens, G.K., 2009. A local search heuristic for the pre-and end-haulage of intermodal container terminals. Comput. Oper. Res. 36 (10), 2763–2772.

Chung, K., Ko, C., Shin, J., Hwang, H., Kim, K., 2007. Development of mathematical models for the container road transportation in korean trucking industries. Comput. Ind. Eng. 53 (2), 252–262.

Clarke, G., Wright, J.W., 1964. Scheduling of vehicles from a central depot to a number of delivery points. Oper. Res. 12 (4), 568–581.

Desroches, M., Lenstra, J.K., Savelsbergh, M.W.P., Soumis, F., 1988. Vehicle routing with time windows: optimization and approximation. Veh. Routing: Methods Stud., 65–84

Erera, A.L., Smilowitz, K.R., 2008. Intermodal drayage routing and scheduling. Int. Freight Transp., Autom. Control Eng. Ser., 171–188

Funke, J., Kopfer, H., 2015. A neighborhood search for a multi-size container transportation problem. 15th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2015, vol. 48(3). IFAC-PapersOnLine, pp. 2041–2046.

Funke, J., Kopfer, H., 2016. Container flows and truck routes in inland container transportation. In: Dynamics in Logistics. Springer, pp. 215–225.

Gavish, B., Srikanth, K., 1986. An optimal solution method for large-scale multiple traveling salesmen problems. Oper. Res. 34 (5), 698–717.

Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. Comput. Oper. Res. 13 (5), 533–549.

Goel, A., Meisel, F., 2013. Workforce routing and scheduling for electricity network maintenance with downtime minimization. Eur. J. Oper. Res. 231 (1), 210–228.

Grötschel, M., 2015. Schnelle Rundreisen: das travelling-salesman-problem. In: Hußmann, S., Lutz-Westphal, B. (Eds.), Diskrete Mathematik erleben. Springer Sepktrum, Wiesbaden, pp. 95–128 (Chapter 4).

Hoffman, K.L., Padberg, M., Rinaldi, G., 2013. Traveling salesman problem. In: Encyclopedia of Operations Research and Management Science. Springer, pp. 1573–1578.

Imai, A., Nishimura, E., Current, J., 2007. A lagrangian relaxation-based heuristic for the vehicle routing with full container load. Eur. J. Oper. Res. 176 (1), 87–105.

Jaw, J.J., Odoni, A.R., Psaraftis, H.N., Wilson, N.H.M., 1986. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. Transp. Res. Part B: Methodol. 20 (3), 243–257.

Jula, H., Chassiakos, A., Ioannou, P., 2006. Port dynamic empty container reuse. Transp. Res. Part E: Logist. Transp. Rev. 42 (1), 43–60.

Jula, H., Dessouky, M., Ioannou, P., Chassiakos, A., 2005. Container movement by trucks in metropolitan networks: modeling and optimization. Transp. Res. Part E: Logist. Transp. Rev. 41 (3), 235–259.

Karp, R.M., 1972. Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W., Bohlinger, J.D. (Eds.), Complexity of Computer Computations, The IBM Research Symposia Series. Springer, US, pp. 85–103.

Konings, R., 2005. Foldable containers to reduce the costs of empty transport? A cost–benefit analysis from a chain and multi-actor perspective. Marit. Econ. Logist. 7 (3), 223–249.

Lai, M., Crainic, T.G., Di Francesco, M., Zuddas, P., 2013. An heuristic search for the routing of heterogeneous trucks with single and double container loads. Transp. Res. Part E: Logist. Transp. Rev. 56, 108–118.

Macharis, C., Bontekoning, Y.M., 2004. Opportunities for or in intermodal freight transport research: a review. Eur. J. Oper. Res. 153, 400–416.

Nossack, J., Pesch, E., 2013. A truck scheduling problem arising in intermodal container transportation. Eur. J. Oper. Res. 230, 666–680.

Notteboom, T.E., Rodrigue, J.-P., 2005. Port regionalization: towards a new phase in port development. Marit. Policy Manage. 32 (3), 297–313.

Parragh, S.N., Doerner, K.F., Hartl, R.F., 2008a. A survey on pickup and delivery problems. Part I: Transportation between customers and depot. J. Betriebswirt. 58 (1), 21–51.

Parragh, S.N., Doerner, K.F., Hartl, R.F., 2008b. A survey on pickup and delivery problems. Part II: Transportation between pickup and delivery locations. J. Betriebswirt. 58 (2), 81–117.

Pesch, E., Glover, F., 1997. TSP ejection chains. Discr. Appl. Math. 76 (1), 165–181.

Popović, D., Vidović, M., Nikolić, M., 2014. The variable neighborhood search heuristic for the containers drayage problem with time windows. In: Soft Computing in Industrial Applications. Springer, pp. 351–364.

Reinelt, G., 1991. TSPLIB—a traveling salesman problem library. ORSA J. Comput. 3 (4), 376–384.

Rosenkrantz, D.J., Stearns, R.E., Lewis II, P.M., 1977. An analysis of several heuristics for the traveling salesman problem. SIAM J. Comput. 6 (3), 563–581.

Savelsbergh, M.W.P., Sol, M., 1995. The general pickup and delivery problem. Transp. Sci. 29 (1), 17–29.

Schönberger, J., Buer, T., Kopfer, H., 2013. A model for the coordination of 20-foot and 40-foot container movements in the hinterland of a container terminal. In: Pacino, D., Voß, S., Jensen, R. (Eds.), ICCL 2013, LNCS, vol. 8197. Springer-Verlag, Berlin Heidelberg, pp. 113–127.

Smilowitz, K., 2006. Multi-resource routing with flexible tasks: an application in drayage operations. Iie Trans. 38 (7), 577–590.

Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. 35 (2), 254–265.

Solomon, M.M., Desrosiers, J., 1988. Survey paper-time window constrained routing and scheduling problems. Transp. Sci. 22 (1), 1–13.

Srour, F.J., Máhr, T., De Weerdt, M.M., Zuidwijk, R.A., 2010. MIPLIB truckload PDPTW instances derived from a real-world drayage case. Tech. Rep., Erasmus Research Institute of Management (ERIM).

Sterzik, S., Kopfer, H., 2013. A tabu search heuristic for the inland container transportation problem. Comput. Oper. Res. 40, 953–962.

Sterzik, S., Kopfer, H., Yun, W.Y., 2012. Reducing hinterland transportation costs through container sharing. Flexible Serv. Manuf. J. 27 (2–3), 382–402.

Van Der Horst, M.R., De Langen, P.W., 2008. Coordination in hinterland transport chains: a major challenge for the seaport community. Marit. Econ. Logist. 10 (1), 108–129.

Vidović, M., Radivojević, G., Raković, B., 2011. Vehicle routing in containers pickup up and delivery processes. Proc. – Soc. Behav. Sci. 20, 335–343.

Vidović, M., Nikolić, M., Popović, D., 2012. Two mathematical formulations for the containers drayage problem with time windows. In: 2nd International Conference on Supply Chains ICSC.

Wang, X., Regan, A.C., 2002. Local truckload pickup and delivery with hard time window constraints. Transp. Res. Part B 36, 97–112.
Williams, H.P., 2013. Model Building in Mathematical Programming. John Wiley & Sons.
Zhang, R., Yun, W.Y., Moon, I., 2009. A reactive tabu search algorithm for the multi-depot container truck transportation problem. Transp. Res. Part E: Logist. Transp. Rev. 45 (6), 904–914.
Zhang, R., Yun, W.Y., Kopfer, H., 2010. Heuristic-based truck scheduling for inland container transportation. OR Spectrum 32, 787–808.
Zhang, R., Yun, W.Y., Moon, I.K., 2011. Modeling and optimization of a container drayage problem with resource constraints. Int. J. Prod. Econ. 133 (1), 351–359.
Zhang, R., Yun, W., Kopfer, H., 2015. Multi-size container transportation by truck: modeling and optimization. Flexible Serv. Manuf. J. 27 (2-3), 403–430.