

A model for the coordination of 20-foot and 40-foot container movements in the hinterland of a container terminal

Jörn Schönberger¹ Tobias Buer^{1,*} Herbert Kopfer¹

A b s t r a c t. Considered is a carrier that requires decision support to organize an efficient transport of loaded and empty containers in the hinterland of a sea port. Loaded containers are handled as pickup-and-delivery requests, however, requests for empty containers are incomplete because either the pickup location of a container or the delivery location of a container is a priori unknown. The problem is modelled as a generalization of the pickup-and-delivery problem (PDP) with less-than-truckload (LTL) requests. Practically speaking, by using LTL request we are able to consider 20-foot and 40-foot containers simultaneously. This is closer to reality than most previous models discussed in the literature which use full truckload requests, i.e., only containers of homogeneous size are possible. Three types of decisions are involved in the proposed model: a determination of pickup or delivery locations for the incomplete requests, a routing of vehicles, and a routing of empty containers. The presented model is validated by means of a numerical example computed by a MIP solver.

Keywords: hinterland container transport, container and request matching, incomplete requests, pickup-and-delivery

¹ University of Bremen, Faculty of Business Studies & Economics, Chair of Logistics, Wilhelm-Herbst-Str. 5, 28359 Bremen, Germany

* corresponding author

This paper is a postprint. The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-642-41019-2_9. Please cite as:

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.M. (eds) Computational Logistics. ICCL 2013. Lecture Notes in Computer Science, vol 8197. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-642-41019-2_9

1 Introduction

Drayage operations involve the transport of containers between a container terminal (sea port or railway yard) and some customer locations in the hinterland of the terminal. Here, the focus is on road-based transport of the containers by means of trucks. The transport requests in drayage operations include movements of loaded as well as empty containers. Usually, containers are transported from customers to terminals or vice versa. The requests are initiated and specified by the customers. Drayage operations are of high importance for the design of efficient intermodal container traffic, because they account for about 40 percent of the total transport costs.

We consider daily drayage operations from the point of view of a trucking company (decision maker) which is responsible to organize the container movements. As new container transport requests arrive over the day the decision maker should apply online planning tools which provide a frequent and reactive plan revision of the needed hinterland container movements. In order to improve planning, the purpose of this paper is to present a decision model for a snapshot of this online planning problem. The model assumes a given set of transport requests together with the current positions of the vehicles and containers. The proposed model is an extension of the well-known less-than-truckload pickup-and-delivery problem (PDP). Unlike previous models for drayage transport operations, the model at hand is able to handle 20-foot and 40-foot containers simultaneously. The model therefore contributes to reduce the gap between real-world drayage planning and mathematical planning models used as core components of decision support systems.

The remaining paper is organized as follows. Section 2 describes the container pickup-and-delivery problem (CPDP) and Section 3 reviews related literature. A model for the CPDP with less-than-truckload requests is presented in Section 4 and is validated by means of a worked numerical example in Section 5. Section 6 concludes the paper.

2 The Container Pickup-and-Delivery Problem

The presented problem is denoted as container pickup-and-delivery problem (CPDP). The CPDP has to be solved by a trucking company that organizes the transport of containers in the hinterland of a port or a rail yard. Fig. 1 shows an exemplary layout of such a hinterland network. There is at least one (e.g. seaside) container terminal, the set of terminal nodes is denoted as $\mathcal{N}^{TERMINAL}$. Through a terminal, containers enter or leave the hinterland. Furthermore, there is set $\mathcal{N}^{CUSTOMER}$ of customer locations. A customer may receive a loaded container from a terminal (*import request*) or ship a loaded container to a terminal (*export request*). At the customer locations there might be an imbalance of inbound and outbound containers. Therefore, some customers have a surplus of *empty* containers while others are in need of empty containers. A customer with a need for an empty container for his or her goods initiates a *provide request*. Hence, the delivery location of a provide request is known but the empty container may be picked up from any location with a surplus of empty containers, e.g., from a container depot or from another customer. On the other hand, a customer with a surplus of empty containers initiates a *storage request* which requires a pick up of an empty container at the customer location, however, the empty container may be delivered to a container depot or to another customer in need of an empty container. Storage and provide requests are also denoted as *incomplete requests*, because either the pickup or the delivery location is initially unknown.

We take the perspective of a trucking company that is responsible for performing all required

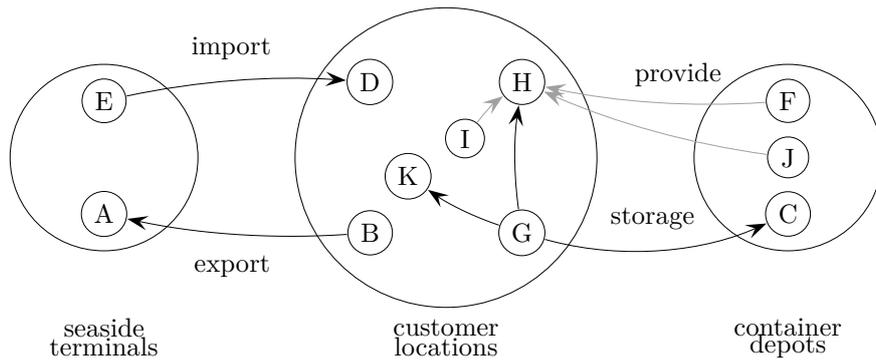


Figure 1: Container flow in the hinterland of a seaport

road-based container movements in the hinterland system necessary to fulfill the customer-specified requests. The considered trucking company maintains a fleet \mathcal{V} of homogenous vehicles that are used to fulfill the container transport requests. The vehicles are located at a single vehicle depot. We assume that it is possible to transport at most two containers – either two 20-foot containers or one 40-foot container – at the same time with a single vehicle. We consider a vehicle as a single entity, i.e., there are no trucks with trailers. However, we assume that a container can be unloaded from a truck at each node of the hinterland network.

There are four types of transport requests (cf. Fig. 1) which are characterized from the point of view of a customer: (i) export requests, (ii) import requests, (iii) storage requests, and (iv) provide requests. Each request r is determined by a triple $(r^+, r^-, c(r))$. This means, that request r requires the transport of a container of size $c(r)$ from the pickup location r^+ to the delivery location r^- . The container size $c(r)$ is measured in twenty-foot equivalent units (TEU) and only $c(r) = 1$ (a 20-foot container) or $c(r) = 2$ (a 40-foot container) are allowed values.

Import and export requests are regular PDP requests, i.e., both the pickup location as well as the delivery location are given. In an import request, a loaded container of known size has to be delivered from a terminal to a customer location, i.e., $(r^+, r^-, c(r)) \in \mathcal{N}^{TERMINAL} \times \mathcal{N}^{CUSTOMER} \times \{1; 2\}$. An export request, on the other hand, requires the transportation of a loaded container of known size from a customer location to a terminal, i.e., $(r^+, r^-, c(r)) \in \mathcal{N}^{CUSTOMER} \times \mathcal{N}^{TERMINAL} \times \{1; 2\}$. We distinguish these two request types for two reasons. On the one hand, this fact may be used in providing tighter model formulations in the future or it may be exploited by a heuristic. This distinction is also used in some related approaches and therefore provides some consistency with the literature, e.g. [14], [6], [12].

By means of a *provide request*, a given customer requires an *empty* container. That is, the delivery location r^- of a provide request r is given. However, the used pickup location is part of the decision problem. We assume, it is always possible to pickup an empty container from a container depot. However, it is also possible to pickup a container from a customer with a surplus of empty containers. Node H in Fig. 1 represents the delivery location of a provide request, i.e., H requires an empty container. A potential pickup location r^+ for the empty container may be the container depot (nodes F or J), or another customer (node I).

By means of a *storage request*, a customer requires the collection of an empty container at a location $r^+ \in \mathcal{N}^{CUSTOMER}$. The collected empty container must be forwarded either to a customer in need for an empty container or to a container depot where the container is stored. Node G in Fig. 1 offers an empty container. The container may be delivered to another customer (e.g. nodes H or K), or to a container depot (node C). Note, we assume that it is

Table 1: Comparison of different models from the literature

	truck- load	incomplete requests	depots	terminals	TW
Jula et al. 2005 [5]	FT	no	1	m	no
Imai et al. 2007 [4]	FT	no	1	1	no
Zhang et al. 2010 [14]	FT	yes	m	m	yes
Braekers et al. 2011 [1]	FT	yes	1	m	yes
Zhang et al. 2011 [13]	FT	yes	1	1	yes
Vidovic et al. 2011 [11]	LTL	no	1	1	no
Reinhardt et al. 2012 [7]	FT	no	1	m	yes
Braekers et al. 2013 [2]	FT	yes	1	m	yes
Wang and Yun 2013 [12]	FT	yes	1	m	yes
Nossack and Pesch 2013 [6]	FT	yes	m	m	yes
CPDP, this paper	LTL	yes	1	1	no

FT = full truckload, LTL = less-than-truckload, m = multiple, TW = time windows

always possible to store a container in a container depot.

The trucking company has to care for three types of decisions. First, store and provide requests have to be completed by selecting an appropriate pickup or delivery location, respectively. Second, for each request a container has to be selected. Third, containers have to be assigned to vehicles and routes for the vehicles have to be generated. The decisions are interdependent and therefore have to be taken into account simultaneously.

Due to the online characteristic of the transportation planning situation it is difficult to state an obvious objective function. However, for the snapshot planning model considered here, the minimization of the transportation effort appears to be a reasonable goal. Therefore and for simplicity, we assume the decision maker wants to minimize the total travel distance of all trucks.

3 Literature Review

A comparison in a nutshell of the literature related to this work is provided by Table 1. The second column states whether the discussed models use full-truckload (FT) or less-than-truckload (LTL) requests. For the scenario at hand, FT requests basically mean that only homogenous container sizes (either 40-foot or 20-foot) are supported. LTL requests support 20-foot and 40-foot containers simultaneously. The third column states whether incomplete requests, i.e., provide and store requests with unknown pickup and delivery locations, respectively, are supported. Column four (five) states whether one or multiple container depots (terminals) are used. The rightmost column indicates, whether time-windows have to be considered for the requests.

As far as we know the approach of [11] is the only approach that considers 20-foot and 40-foot containers simultaneously. However, the problem is modeled as a variant of a vehicle routing problem with backhauls and excludes the possibility that storage and provide requests are performed in the same tour, i.e., empty containers may only be transported from and to the terminal but not to customer locations which limits the potential to reduce transportation costs.

The paper [5] was among the first to consider drayage scenarios. The proposed model considers container transport between customers and a terminal and models this situation as a multiple traveling salesman problem with time windows. Incomplete store and provide requests are not supported, however.

Likewise, [4] study the vehicle routing problem with full container load with a priori known pickup and delivery locations of all requests. A container is permanently assigned to a vehicle. That is, a vehicle which fulfills an import request first and then an export requests uses for both requests the same container. This is not required in our formulation. Furthermore, an inhomogeneous fleet regarding travel cost and maximum travel time, but not with respect to the supported container sizes, is considered. In [3] the vehicle routing with full containers (VRPFC) is extended by time windows and solved as a VRPTW. The paper [7] extends the model proposed in [3] by multiple container terminals. The approach is tested by means of real world data.

[13] model the problem with flexible tasks. A flexible tasks is a request with an a priori unknown delivery or pickup location which is equal to our approach. Furthermore, they consider a dynamic planning situation which allows that requests may be added during the day.

In contrast to the previously mentioned approaches, empty containers are explicitly considered as transportable resources in Zhang et al. [14]. The model is denoted as Inland Container Transportation Problem (ICT) and takes into account multiple terminals and multiple depots. The ICT is also studied in [9] where a solution approach based on tabu search is proposed. In [10] a variant of the ICT model is used to evaluate the potential of collaborations between the several carriers in the hinterland. [12] extend the ICT by considering train transportation in addition to truck transportation. The model of [6] also builds upon the model of Zhang et al. [14]. Each truck serves exactly one container (i.e. only 40-foot containers are regarded). Consolidation of two 20-foot containers on a truck does not take place.

In a recent paper [2] also considers the situation where pickup or delivery locations of empty containers are undefined. An empty container allocation problem is proposed which is based on a the transportation problem before solving the routing problem. However, only homogenous container sizes are considered. The goal is to minimize the travel distance of empty containers

All in all, the CPDP presented here distinguishes itself from previous models by explicitly taking into account 20-foot and 40-foot containers simultaneously and supporting store and provide requests with unknown pickup or delivery locations. However, in contrast to many models from the literature time windows are not yet considered.

4 Mixed-Integer Linear Program

The parameters of CPDP and the decision variables are introduced in Section 4.1. Section 4.2 presents the constraints and the objective function.

4.1 Parameters and Decision Variables

The basic idea of our decision model is to integrate the three decision tasks request completion, container flow determination and vehicle routes into one mixed-integer linear program. We deploy a first set of decision variables which represents the request completion decision problem. Another set of decision variables controls the container flow and a third decision variable set is dedicated to the compilation of executable vehicle routes. Two additional decision variable families are introduced in order to control the assignment of completed requests to containers

Table 2: Decision Variables

completion of requests	
$z_{ik}^+ \in \{0; 1\}$	equals 1 if and only if node i is the pickup node of container k
$z_{ik}^- \in \{0; 1\}$	equals 1 if and only if node i is the delivery node of container k
assignment of requests to containers	
$y_{rk} \in \{0; 1\}$	equals 1 if and only if request r is assigned to container k
determination of the container flow	
$x_{ijk}^{CON} \in \{0; 1\}$	equals 1 if and only if container k flows along arc (i, j)
$t_{ik}^{CON} \in \mathbb{R}^{\geq 0}$	visiting time of container k at node i
assignment of containers to vehicles	
$w_{kf} \in \{0; 1\}$	equals 1 if and only if container k is assigned to vehicle f
determination of the vehicle routes	
$x_{ijf}^{VEH} \in \{0; 1\}$	equals 1 if and only if vehicle f flows along arc (i, j)
$t_{if}^{VEH} \in \mathbb{R}^{\geq 0}$	visiting time of vehicle f at node i
$l_{if} \in \mathbb{N}$	outbound payload of vehicle f leaving node i
$\delta_{if} \in \mathbb{Z}$	payload variation of vehicle f at node i

and for conducting the matching of the flows of containers and the vehicle routes. All five decision variable collections are declared in Tab. 2.

Although we consider only one seaside / railroad terminal, we introduce a terminal node for each single container in the system, e.g. if the number of containers in the system is 5 then nodes 1 to 5 are nodes that belong to the terminal and node i can be used exclusively by container i . We use this node-duplication strategy in order to make the control of the in- and outbound flow of containers into the terminal easier. We apply the same concept for only the container depot, e.g. there are $N^{CONTAINER}$ nodes that form the container depot but each node can be visited or left by exactly one container. Since a vehicle is allowed to visit the terminal (or the container depot) several times it is not necessary that we store different arrival times of a vehicle at the depot. Instead we need to store only one arrival time of a vehicle at a node. We are going to use the recursive arrival time update [8] for short cycle prevention in the vehicle flows. Therefore, we need a common outbound node and a common inbound node for each vehicle depot.

The set of all involved nodes is \mathcal{N} and \mathcal{N}^{VDEPOT} is the set of all nodes belonging to the truck depot. Set \mathcal{V} is the set of vehicles and set \mathcal{C} is the set of all containers. The set of all requests is $REQS$. It is partitioned into four subsets, i.e., the set $REQS^{EXP}$ of export requests, the set $REQS^{IMP}$ of import requests, the set $REQS^{STOR}$ of storage requests, and the set $REQS^{PROV}$ of provide requests.

The binary parameter $AVAIL(k, i)$ equals 1 if and only if container k is available at node i . Similarly, the binary parameter $REQUI(k, i)$ equals 1 if and only if container k is allowed to be delivered to node i . The length of arc (i, j) is $d(i, j)$. A rout of vehicle f starts at node $START(f)$ and ends in node $STOP(f)$. If the pickup (delivery) node of request r is specified by the customer then this node is referred to as $REQSTART(r)$ ($REQSTOP(r)$). The current position of container k is $CONT_LOC(k)$ and the binary parameter $FIXED(k, r)$ equals 1 if and only if request r is already matched with container k (this is true for all import, export

and storage requests). The size of container k is given by $C(k)$. A 20-foot container has size $C(k) = 1$ and a 40-foot container has size $C(k) = 2$. Finally, \mathbf{M} is a sufficiently large integer number.

We construct a complete directed graph from the node set \mathcal{N} , which means that the arc set of the graph equals $\mathcal{N} \times \mathcal{N}$. The arcs are weighted by their travel distance $d(i, j)$. We assume that all vehicles travel at unit speed so that the absolute value of $d(i, j)$ equals the absolute value of the time needed to travel along arc (i, j) .

4.2 Constraints and Objective Function

In the following, we present the constraints required to ensure the feasibility of the generated vehicle routes. The selection of loading and unloading locations for all containers is controlled by constraints (1)–(16). The assignment of requests to containers is determined by constraints (17)–(20). Container flows are controlled by the constraints (21)–(26), and the generation of vehicle routes is subject to constraints (27)–(47). The matching of container flows and vehicle routes is achieved by constraints (48)–(54).

4.2.1 Selection of Locations for Loading and Unloading

$$z_{CONT_LOC(k)k}^+ \geq \sum_{r \in REQS} y_{rk} \quad \forall k \in \mathcal{C} \quad (1)$$

$$\sum_{i \in \mathcal{N}} z_{ik}^+ \geq y_{rk} \quad \forall k \in \mathcal{C}, r \in REQS \quad (2)$$

$$\sum_{i \in \mathcal{N}} z_{ik}^- \geq y_{rk} \quad \forall k \in \mathcal{C}, r \in REQS \quad (3)$$

$$\sum_{k \in \mathcal{C}} z_{ik}^+ \leq 1 \quad \forall i \in \mathcal{N} \quad (4)$$

$$\sum_{k \in \mathcal{C}} z_{ik}^- \leq 1 \quad \forall i \in \mathcal{N} \quad (5)$$

The start node for each container is known (1). If request r is assigned to container k then a start node must be selected for container k (2). Similarly, a stop node is selected for container k assigned to request r (3). A node can become the starting point of at most one container (4). Similarly, it is not allowed that two or more containers are delivered to a node (5).

$$z_{REQSTART(r)k}^+ \geq y_{rk} \quad \forall r \in REQS^{EXP}, k \in \mathcal{C} \quad (6)$$

$$z_{REQSTOP(r)k}^- \geq y_{rk} \quad \forall r \in REQS^{EXP}, k \in \mathcal{C} \quad (7)$$

$$z_{REQSTART(r)k}^+ \geq y_{rk} \quad \forall r \in REQS^{IMP}, k \in \mathcal{C} \quad (8)$$

$$z_{REQSTOP(r)k}^- \geq y_{rk} \quad \forall r \in REQS^{IMP}, k \in \mathcal{C} \quad (9)$$

$$z_{REQSTART(r)k}^+ \geq y_{rk} \quad \forall r \in REQS^{STOR}, k \in \mathcal{C} \quad (10)$$

$$z_{REQSTOP(r)k}^- \geq y_{rk} \quad \forall r \in REQS^{PROV}, k \in \mathcal{C} \quad (11)$$

The constraint (6) ((7)) assigns the pickup (delivery) node of the export request r to container k . For each import request, the two constraints (8) and (9) determine the corresponding container pickup and the container delivery location. The start node for each container being assigned to a storage request is known (10) but for each provision request only the destination node is

known (11).

$$\sum_{i \in \mathcal{N} | REQUI(k,i) > 0} z_{ik}^- \geq y_{rk} \quad \forall r \in REQS^{STOR}, k \in \mathcal{C} \quad (12)$$

$$\sum_{i \in \mathcal{N} | AVAIL(k,i) > 0} z_{ik}^+ \geq y_{rk} \quad \forall r \in REQS^{PROV}, k \in \mathcal{C} \quad (13)$$

$$z_{ik}^- \leq REQUI(k, i) \quad \forall i \in \mathcal{N}, k \in \mathcal{C} \quad (14)$$

$$z_{ik}^+ \leq AVAIL(k, i) \quad \forall i \in \mathcal{N}, k \in \mathcal{C} \quad (15)$$

$$z_{ik}^+ + z_{ik}^- \leq 1 \quad \forall i \in \mathcal{N}, k \in \mathcal{C} \quad (16)$$

Constraint (12) ensures that only a node that needs a container becomes the terminating node of a storage request. Constraint (13) ensures that only a node that stores a container becomes the start node of a provision request. No node receives more containers than requested (14) and no node provides more containers than available at this node (15). The starting and the terminating node assigned to a container are unequal (16).

4.2.2 Request to Container Assignment

$$\sum_{k \in \mathcal{C}} y_{rk} = 1 \quad \forall r \in REQS \quad (17)$$

$$\sum_{r \in REQS} y_{rk} \leq 1 \quad \forall k \in \mathcal{C} \quad (18)$$

$$y_{rk} = FIXED(k, r) \quad \forall r \in REQ^{EXP}, k \in \mathcal{C} \quad (19)$$

$$y_{rk} = FIXED(k, r) \quad \forall r \in REQ^{IMP}, k \in \mathcal{C} \quad (20)$$

Each request is assigned to exactly one container (17) and each container receives at most one request (18). The two constraints (19) and (20) ensure consistency of fully known import and export requests with the corresponding used containers.

4.2.3 Container Flows

$$x_{iik}^{CON} = 0 \quad \forall i \in \mathcal{N}, k \in \mathcal{C} \quad (21)$$

$$\sum_{j \in \mathcal{N} | j \neq i} x_{ijk}^{CON} \geq z_{ik}^+ \quad \forall i \in \mathcal{N}, k \in \mathcal{C} \quad (22)$$

$$\sum_{j \in \mathcal{N} | j \neq i} x_{jik}^{CON} \geq z_{ik}^- \quad \forall i \in \mathcal{N}, k \in \mathcal{C} \quad (23)$$

$$z_{jk}^+ + \sum_{i \in \mathcal{N}} x_{ijk}^{CON} = z_{jk}^- + \sum_{i \in \mathcal{N}} x_{jik}^{CON} \quad \forall j \in \mathcal{N}, k \in \mathcal{C} \quad (24)$$

Self-referencing flows of containers are not allowed (21). If node i has been selected as starting node of the flow of container k then container k leaves node i (22). In case that i has been selected to be the terminal node of the flow of container k then constraint (23) ensures that k travels to i . Constraint (24) preserves the balance of inbound and outbound flow of container k at node i . If i is the selected start node of the flow of container k then there is no inbound flow of k into i . Similarly, if i is selected to be the terminus node of container k then there is no outbound flow of k from i .

$$t_{ik}^{CON} \geq 0 \quad \forall i \in \mathcal{N}, k \in \mathcal{C} \quad (25)$$

$$t_{ik}^{CON} + d(i, j) \cdot x_{ijk}^{CON} \leq t_{jk}^{CON} + (1 - x_{ijk}^{CON}) \cdot \mathbf{M} \quad \forall i, j \in \mathcal{N}, k \in \mathcal{C} \quad (26)$$

4.2.4 Vehicle Routing

In order to avoid short cycles of the flowing containers, we recursively calculate the arrival times of container k at the nodes along the determined flow path (25)-(26) as proposed by [8].

$$x_{ijf}^{VEH} = 0 \quad \forall f \in \mathcal{V}, i \in \mathcal{N} \quad (27)$$

$$\sum_{j \in \mathcal{N}} x_{START(f),j,f}^{VEH} = 1 \quad \forall f \in \mathcal{V} \quad (28)$$

$$\sum_{j \in \mathcal{N}} x_{j,STOP(f),f}^{VEH} = 1 \quad \forall f \in \mathcal{V} \quad (29)$$

$$\sum_{j \in \mathcal{N}} x_{jif}^{VEH} = \sum_{j \in \mathcal{N}} x_{ijf}^{VEH} \quad \forall f \in \mathcal{V}, i \in \mathcal{N} \setminus \mathcal{N}^{VDEPOT} \quad (30)$$

Self-referencing loops are forbidden by (27). Each vehicle leaves its home position (28) and terminates in its designated terminus node (29). Constraint (30) ensures the vehicle flow preservation at nodes away from the vehicle depot.

$$t_{if}^{VEH} \geq 0 \quad \forall i \in \mathcal{N}, f \in \mathcal{V} \quad (31)$$

$$t_{if}^{VEH} + d(i, j) \cdot x_{ijf}^{VEH} \quad (32)$$

$$\leq t_{jf}^{VEH} + (1 - x_{ijf}^{VEH}) \cdot \mathbf{M} \quad \forall i, j \in \mathcal{N}, f \in \mathcal{V} \quad (33)$$

$$t_{if}^{VEH} \leq t_{jf}^{VEH} + (1 - w_{kf}) \cdot \mathbf{M} \quad (34)$$

$$+(1 - z_{ik}^+) \cdot \mathbf{M} + (1 - z_{jk}^-) \cdot \mathbf{M} \quad \forall i, j \in \mathcal{N}, f \in \mathcal{V}, k \in \mathcal{C} \quad (35)$$

We use the arrival time - based prevention of short cycles of the vehicle routes (31)-(33). Constraint (35) guarantees that vehicle f visits the pickup location i of container k (equiv. to $z_{ik}^+ = 1$) earlier than the associated delivery location j of k (equiv. to $z_{jk}^- = 1$) if vehicle f has been selected to move container k (equiv. to $w_{kf} = 1$).

$$t_{if}^{VEH} = 0 \quad \forall i \in \mathcal{N}, f \in \mathcal{V} \quad (36)$$

$$t_{STOP(f),f}^{VEH} \leq T_{max} \quad \forall f \in \mathcal{V} \quad (37)$$

All vehicles depart from their depot at time 0 (36) and return there not later than T_{max} (37).

$$l_{START(f),f} = 0 \quad \forall f \in \mathcal{V} \quad (38)$$

$$l_{STOP(f),f} = 0 \quad \forall f \in \mathcal{V} \quad (39)$$

$$\delta_{START(f),f} = 0 \quad \forall f \in \mathcal{V} \quad (40)$$

$$\delta_{STOP(f),f} = 0 \quad \forall f \in \mathcal{V} \quad (41)$$

Each vehicle leaves its starting node empty (38) and heads emptied towards its terminus node (39). No load variation is effective at these two locations (40),(41).

$$\delta_{if} + (1 - w_{kf}) \cdot \mathbf{M} + (1 - z_{ik}^+) \cdot \mathbf{M} \geq C(k) \quad \forall i \in \mathcal{N}, k \in \mathcal{C}, f \in \mathcal{V} \quad (42)$$

$$\delta_{if} \leq C(k) + (1 - z_{ik}^+) \cdot \mathbf{M} \quad \forall f \in \mathcal{V}, i \in \mathcal{N}, k \in \mathcal{C} \quad (43)$$

$$\delta_{if} + (1 - w_{kf}) \cdot \mathbf{M} + (1 - z_{ik}^-) \cdot \mathbf{M} \leq -C(k) \quad \forall i \in \mathcal{N}, k \in \mathcal{C}, f \in \mathcal{V} \quad (44)$$

$$\delta_{if} \geq -C(k) - (1 - z_{ik}^-) \cdot \mathbf{M} \quad \forall f \in \mathcal{V}, i \in \mathcal{N}, k \in \mathcal{C} \quad (45)$$

We set, by constraints (42) and (43), the payload variation δ_{if} of vehicle f at node i to $C(k)$ if container k has been assigned to vehicle f ($w_{kf} = 1$) and if i is the starting (pickup) node

associated with container k ($z_{ik}^+ = 1$). Similarly, we set, by constraints (44) and (45), the payload variation δ_{if} of vehicle f at node i to $-C(k)$ if container k has been assigned to vehicle f ($w_{kf} = 1$) and if i is the terminus (delivery) node associated with container k ($z_{ik}^- = 1$).

$$l_{if} + \delta_{if} \leq l_{jf} + (1 - x_{ijf}^{VEH}) \cdot \mathbf{M} \quad \forall i, j \in \mathcal{N}, f \in \mathcal{V} \quad (46)$$

$$l_{jf} \leq 2 + (1 - \sum_{i \in \mathcal{N}} x_{jif}^{VEH}) \cdot \mathbf{M} \quad \forall j \in \mathcal{N}, f \in \mathcal{V} \quad (47)$$

The payload variation along the route of vehicle f is update by δ_{if} recursively (46) so that the payload is increased by at least 1 (TEU) if f visits a node with a container pickup but the payload is reduced by at least 1 (TEU) if f visits a container delivery node. It is not allowed that a vehicle carries a payload of more than 2 TEU at any position of its route (47).

4.2.5 Matching of Container Flows and Vehicle Routes

$$\sum_{f \in \mathcal{V}} w_{cf} \leq 1 \quad \forall c \in \mathcal{C} \quad (48)$$

$$\sum_{f \in \mathcal{V}} w_{cf} \geq \sum_{r \in REQ_S} y_{rc} \quad \forall c \in \mathcal{C} \quad (49)$$

Each container is assigned to at most one vehicle for being moved (48). If container k is assigned to a request r then it is necessary that k is assigned to exactly one vehicle f (49).

$$x_{ijc}^{CON} \leq x_{ijf}^{VEH} + (1 - w_{cf}) \cdot \mathbf{M} \quad \forall i, j \in \mathcal{N}, c \in \mathcal{C}, f \in \mathcal{V} \quad (50)$$

In the situation where container k moves along arc (i, j) and if container k has been assigned to vehicle f it is necessary that vehicle f also travels along arc (i, j) (50).

$$2 \cdot \left(\sum_{k \in \mathcal{C}} w_{kf} \right) \geq -1 + \sum_{i, j \in \mathcal{N}} x_{ijf}^{VEH} \quad \forall f \in \mathcal{V} \quad (51)$$

With the goal to avoid the insertion of dummy nodes into the route of vehicle f with the goal to artificially reduce the payload of v , we restrict the number of visited nodes to the least possible number (51), e.g. if m requests are assigned to f then the least necessary number of visited nodes is $2m + 1$ (the starting node is not visited but only left by f).

$$\sum_{r \in REQ_S} y_{rk} = \sum_{f \in \mathcal{V}} w_{kf} \quad \forall k \in \mathcal{C} \quad (52)$$

In order to prevent the incorporation of nodes associated with container that have not been selected for incorporation into the routes of the vehicles constraint (52) is added to the model.

$$\sum_{r \in REQ_S} y_{rk} = \sum_{i \in \mathcal{N}} z_{ik}^- \quad \forall k \in \mathcal{C} \quad (53)$$

$$\sum_{r \in REQ_S} y_{rk} = \sum_{i \in \mathcal{N}} z_{ik}^+ \quad \forall k \in \mathcal{C} \quad (54)$$

Finally, constraints (53) and (54) are used to prevent the visitation of containers for which a pickup and a delivery node are specified but which is not assigned to any request.

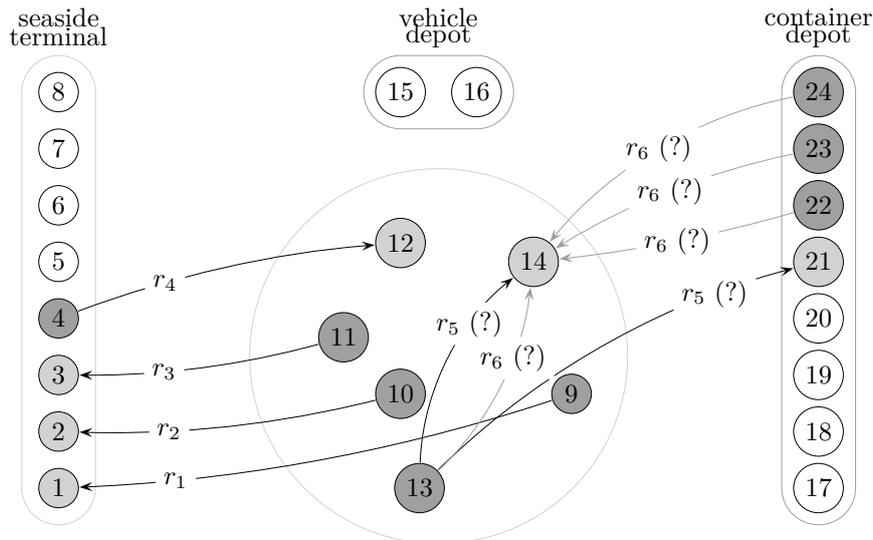


Figure 2: Example scenario with 6 requests, 8 containers, and 5 vehicles. Request r_5 and r_6 are a priori incomplete.

4.2.6 Objective Function

$$\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \sum_{f \in \mathcal{V}} d(i, j) \cdot x_{ijf}^{VEH} \rightarrow \min \quad (55)$$

The trucking company aims at minimizing the overall travel distance of their vehicles (55).

5 Numerical Example

The presented decision model is very difficult to solve. Using appropriate parameter instantiations, well-studied NP-hard decision problems like the pickup and delivery problem are identified as special cases of the presented model. We cannot expect to solve this model to optimality within an acceptable time. Initial computational experiments with the commercial solver CPLEX showed that even quite small instances of the integrated request completion, container flow and vehicle routing problem cannot be solved within reasonable computation time independently from the applied objective function. Often, the solver even failed to identify a feasible solution.

Being aware of the challenging difficulty of the CPDP, we focus on the assessment of the suitability of the presented constraint collection for a rather small example scenario. We instruct CPLEX to identify any feasible solution and terminate the search then. We use the identified solution to verify the feasibility with respect to the aforementioned problem constraints. In order to get further insights into the performance of the model it is necessary to develop specific (meta-)heuristics which is however beyond the scope of the present paper.

The example shown in Fig. 2 comprises a hinterland system with eight 20-foot containers. There is a (seaside or railway) container terminal comprising the nodes 1 to 8. Six customer locations are represented by nodes 9 to 14. Two trucks of capacity 2 TEU are available at a vehicle depot with outbound node 15 and inbound node 16. One hinterland container depot (nodes 17 to 24) can be used to store empty containers or to pickup empty containers needed for the fulfillment of a provide request. Nodes with a surplus of containers are shaded dark

determined for those customers that demand an empty container. The proposed model for the container pickup-and-delivery problem (CPDP) deals with these additional decisions simultaneously and integrated with a less-than-truckload PDP model. For the first time, transport of 20-foot and 40-foot containers in drayage operations can be managed by means of a single model. Future research will focus on the development of heuristic solution approaches for the CPDP.

References

- [1] Braekers, K., Caris, A., Janssens, G.K.: A deterministic annealing algorithm for a bi-objective full truckload vehicle routing problem in drayage operations. *Procedia - Social and Behavioral Sciences* 20, 344 – 353 (2011)
- [2] Braekers, K., Caris, A., Janssens, G.: Integrated planning of loaded and empty container movements. *OR Spectrum* 35(2), 457–478 (2013)
- [3] Caris, A., Janssens, G.: A local search heuristic for the pre- and end-haulage of intermodal container terminals. *Computers & Operations Research* 36(10), 2763 – 2772 (2009)
- [4] Imai, A., Nishimura, E., Current, J.: A lagrangian relaxation-based heuristic for the vehicle routing with full container load. *European Journal of Operational Research* 176(1), 87 – 105 (2007)
- [5] Jula, H., Dessouky, M., Ioannou, P., Chassiakos, A.: Container movement by trucks in metropolitan networks: modeling and optimization. *Transportation Research Part E: Logistics and Transportation Review* 41(3), 235 – 259 (2005)
- [6] Nossack, J., Pesch, E.: A truck scheduling problem arising in intermodal container transportation. *European Journal of Operational Research* (2013), DOI: 10.1016/j.ejor.2013.04.042
- [7] Reinhardt, L., Spoorendonk, S., Pisinger, D.: Solving vehicle routing with full container load and time windows. In: Hu, H., Shi, X., Stahlbock, R., Voß, S. (eds.) *Computational Logistics, LNCS*, vol.7555, pp. 120–128. Springer Berlin Heidelberg (2012)
- [8] Sigurd, M., Pisinger, D., Sig, M.: The pickup and delivery problem with time windows and precedences. Tech. Rep. Technical Report 00/08, DIKU, University of Copenhagen (2008)
- [9] Sterzik, S., Kopfer, H.: A tabu search heuristic for the inland container transportation problem. *Computers & Operations Research* 40(4), 953 – 962 (2013)
- [10] Sterzik, S., Kopfer, H., Yun, W.Y.: Reducing hinterland transportation costs through container sharing. *Flexible Services and Manufacturing Journal* (2012), DOI: 10.1007/s10696-012-9167-y
- [11] Vidovic, M., Radivojevic, G., Rakovic, B.: Vehicle routing in containers pickup up and delivery processes. *Procedia - Social and Behavioral Sciences* 20, 335 – 343 (2011)
- [12] Wang, W.F., Yun, W.Y.: Scheduling for inland container truck and train transportation. *International Journal of Production Economics* 143(2), 349 – 356 (2013)

- [13] Zhang, G., Smilowitz, K., Erera, A.: Dynamic planning for urban drayage operations. *Transportation Research Part E: Logistics and Transportation Review* 47(5), 764 – 777 (2011)
- [14] Zhang, R., Yun, W., Kopfer, H.: Heuristic-based truck scheduling for inland container transportation. *OR Spectrum* 32, 787–808 (2010)