

A column generation based decomposition and aggregation approach for combining orders in inland transportation of containers

Xinan Yang*, Hajem A. Daham[†]

Abstract

A significant portion of the total cost of the intermodal transportation is generated from the inland transportation of containers. In this paper, we design a Mixed Integer Linear Programming (MILP) model for combining orders in the inland, haulage transportation of containers. The pickup and delivery process of both 20 and 40 foot containers from the terminals to the customer locations and vice versa are optimized using heterogeneous fleet consisting of both 20ft and 40ft trucks/chasses. Important operational constraints such as the time window at order receivers, the payload weight of containers and the regulation of the working hours are considered. Based on an assignment problem structure, this MILP solves efficiently to optimality for problems with up to 120 orders.

To deal with larger instances, a decomposition and aggregation heuristic is designed. The basic idea of this approach is to decompose order locations geographically into fan-shaped sub-areas based on the angle of the order location to the port baseline, and solve the sub problems using the proposed MILP model. To balance the fleet size amongst all subgroups, column generation is used to iteratively adjust the number of allocated trucks according to the shadow-price of each truck type. Based on decomposed solutions, orders that are "fully" combined with others are removed and an aggregation phase follows to enable wider combination choices across subgroups. The decomposition and aggregation solution process is tested to be both efficient and cost-saving.

Keywords: Combining Orders, Inland Transportation, MILP Model, Heuristic Decomposition, Column generation.

1 Introduction

The intermodal freight transportation is defined as the delivery of products from origins to destinations in containers using multiple modes of transportation. Inland transportation,

*Dept. of Mathematical Sciences, University of Essex, Colchester CO4 3SQ, United Kingdom. email: xyangk@essex.ac.uk, Telephone: +44(0)1206 872787

[†]Al Muthanna University, Samawah, Iraq. email: hajem.daham@mu.edu.iq, Telephone: +9647716451697

which comprises of rail, trucks and barge transportation modes, is an important part of the intermodal freight transportation. According to the United Nations Conference on Trade and Development (UNCTAD (2015)), around 80% of the global trade volume is transported by sea and handled by ports. Consequently, the demand of transporting containers between the port terminal and its customers is high and increasing (Crainic and Kim (2006)). Comparing to the sea transportation, inland delivery (which is also called drayage) of containers by rail and trucks involves shorter distances but more significant costs, which was estimated between 40-80% of the total cost (Notteboom and Rodrigue (2005)). Konings (2005) also reported that 40% of the total cost of the intermodal transportation was generated by road transportation. Considering the large volume of inland container transportation demand and how costly it is, good management strategies which are efficient enough to be suitable for large industrial implementations are needed (Morlok and Spasovic (1994)). The nature of the Road Container Transportation (RCT) makes it a type of the Pick-up and Delivery Problems (PDP). Parragh et al. (2008a,b) provided an extensive classification and explanation of the PDP types. However, the RCT is an extension from these types since it comprises the pick-up and delivery of different types of containers (loaded and empty) between customers, the port and inland depots (Lun and Cheng (2010)).

In this paper, we design a Mixed Integer Linear Programming (MILP) model for the combination of orders in the truck transportation of containers. In this model, the pick-up and delivery of both 20 and 40 foot containers between the port terminal and customer locations are considered. The model is an extension of Hajem et al. (2017), which considers only 20ft, fully loaded shipments. We expand the previous work by considering heterogeneous truck types and container sizes (20ft and 40ft), and by allowing the simultaneous transportation of two 20ft containers on a 40ft truck chassis. All other practical restrictions remain, such as the time window at the port and customer locations, the weight restrictions for single and double chassis types, the penalty cost for the potential overtime working of truck drivers, etc. Based on the same assignment structure of Hajem et al. (2017), this model solves efficiently the problem in question; examples with up to 120 orders can be solved by CPLEX, which generates the optimal delivery plan satisfying all constraints in just a few minutes. Note that empty containers are not explicitly considered in this model, however many of them can also be covered by the model and the proposed approach, given they have at least one end of the (origin, destination) at the port (the other end can be either an inland depot or a customer location). In other words, street-turn of empty containers is not allowed. This restricts the proposed model from considering more than four containers per route. However, this work is still useful in case: (a) All loaded imports return their container to the port, after delivery at the customer; (b) All loaded exports are serviced by driving an empty container from the port, before collection at the customer. These are standard operational rules implemented by many shipping lines in the market.

In order to deal with larger instances, a decomposition and aggregation heuristic is designed. The basic idea of this approach is to decompose the locations of orders geographically into small subgroups and solve the subgroups problems by the formulated MILP, to obtain the optimal order-combination plan within each subgroup. Decomposi-

tion is proposed based on the angles of the customer location to the port baseline, which creates fan-shaped subareas which allows all routes to start from/finish at the port without leaving the subarea. To balance fleet sizes amongst all subgroups, column generation is used to iteratively adjust the number of allocated trucks according to the shadow-price of each truck type. The decomposed model is smaller in size and solves efficiently, but the result is not optimal since we prevent some combination choices through decomposition. Therefore in the second phase, we do aggregation by removing the “well-combined” orders, i.e. the order combinations that have already used the full truck load, so as to reduce the problem size. A new MILP is constructed for the aggregated data with remaining orders, and solved as usual. This decomposition and aggregation approach is justified efficient and cost-saving through intensive numerical experiments on real data.

The paper is structured as follows. A literature review is carried out in Section 2. In Section 3 the problem description and the optimisation model are described. The heuristic approach is demonstrated with an example in Section 4. Numerical experiments are presented in Section 5. Section 6 draws conclusions.

2 Literature Review

Many studies have been established to consider the transportation of containers and the majority of them are focusing on the usage of Operational Research techniques. A wide range of descriptions and classifications were introduced in a survey by Steenken et al (2004), which was later expanded and updated by Stahlbock and VoB (2008). More specifically, Braekers et al. (2011) presented an overview of empty container management by examining into three different planning levels (strategic, tactical and operational). In the following, we review the literature on the combination of heterogeneous container types in road transportations and position our work.

The haulage transportation of containers normally refers to the delivery of both of 20ft and 40ft containers using homogenous or heterogeneous fleets, where 40ft trucks can carry up to two 20ft containers or one 40ft container at a time. This type of problem suffers from very limited capacity (at most two containers per truck) and long travel distances. Under this broad category, some literature aim to minimize the transportation cost/travelling distance (Table 1) while some others consider the fleet management aiming to minimize the number of required fleet or to maximize fleet utilization (Table 2).

Likewise to our problem settings focusing on cost-minimization, literature can further be split into delivery management of homogenous and heterogeneous sized containers, and/or according to the type of loading/unloading strategies, i.e. *strip* and *discharge*. In the *strip* case, container, together with its cargo (if not empty), is put onto/removed from the carrying truck at customer locations. The customer then deals with the container by loading/removing its cargo and makes another request for the pick-up/removal of empty/loaded containers. While in the *discharge* case, only the cargo will be loaded onto/unloaded from the delivery truck; containers and trucks are not separated after services. Our study, like most of the works in Table 1, consider *strip* of containers. While

Literature	Strip	Discharge	Type of Containers	Methodology
Chung et al. (2007)	✓	x	heterogeneous	Exact
Zhang et al. (2015)	✓	x	heterogeneous	Exact & Heuristic
Vidović et al. (2011)	✓	x	heterogeneous	Exact & Heuristic
Vidović (2012)	✓	x	heterogeneous	Exact
Lai (2013)	x	✓	heterogeneous	Exact & Heuristic
Lai et al. (2013)	x	✓	heterogeneous	Exact & Heuristic
Wen and Zhou (2007)	✓	x	heterogeneous	Exact & Heuristic
Popović et al. (2014)	✓	x	heterogeneous	Exact & Heuristic
Reinhardt and Pisinger (2012)	x	✓	homogenous	Exact
Reinhardt et al. (2016)	x	✓	homogenous	Exact
Funke and Kopfer (2015, 2016)	✓	✓	heterogeneous	Exact & Heuristic
Hajem et al. (2017)	x	✓	homogenous	Exact
Nordsieck and Schönberger (line)	✓	x	heterogeneous	Exact & Heuristic
This article	✓	x	heterogeneous	Exact & Heuristic

Table 1: Literature considering minimization of delivery costs

Literature	Strip	Discharge	Type of Containers	Methodology
Wang and Regan (2002)	✓	x	homogenous	Exact & Heuristic
Gronalt et al. (2003)	x	✓	homogenous	Exact & Heuristic
Chung et al. (2006)	✓	x	homogenous	Exact & Heuristic
Smilowitz (2006)	x	✓	homogenous	Exact
Chung et al. (2008)	x	✓	homogenous	Exact & Heuristic
Namboothiri and Erera (2008)	✓	x	homogenous	Exact & Heuristic
Shiri and Huynh (2016, 2018)	✓	x	homogenous	Exact & Heuristic

Table 2: Literature considering fleet management

distinguished from most of them which are based on multiple Travelling Salesman Problem (m -TSP) or Vehicle Routing Problems (VRP) models, our work develops an assignment mixed integer linear programming (MILP) model to deal with the same situation. This model solves much faster than the VRP by pre-defining all feasible delivery routes, which improves the current exact methods by solving larger sized examples within reasonable time and memory consumption, as well as allowing a decomposition based heuristic to solve large size instances. Comparing to the typical route-construction heuristics as used by other works, decomposition is more intuitive and widely used in industry as it requires definitely less efforts to implement.

On the other hand, many literatures studied the inland transportation of containers from the prospective of fleet management as summarised in Table 2. In this case, the focus is put on maximizing the truckload usage rather than/on top of travelling distances. Our article is not directly aiming to maximize fleet utilization, however, fleet management ideas are implicitly investigated in the decomposition step and a column generation approach is designed to balance fleet sizes/usages in different subareas.

Another article which is closely related to this one, but in a slightly different context as it addresses the time windowed railroad engine scheduling problem, is Lübbecke and Zimmermann (2003). Despite the different application area, the modelling and solution methodologies of Lübbecke and Zimmermann (2003) share certain similarities with this article due to the pickup and delivery nature of both problems. In modelling, Lübbecke and Zimmermann (2003) also classifies the possible delivery sequences (named as “patterns”) beforehand and proposes an MILP to allocate the pattern(s) to engines. However,

instead of route configurations of the whole route as proposed in our article, Lübbecke and Zimmermann (2003) uses route segments, treats them as virtual nodes in a standard VRP-TW and uses the MILP model to join them together while allocating the best engine to execute the routes. Column generation is also used by Lübbecke and Zimmermann (2003) to solve the problem, with a RMP deciding which subset of orders are served by which engine and subproblems informing the optimal routes to fulfil the allocated orders. Instead of allocating orders (patterns) to trucks as set-partitioning, our column generation approach is applied jointly with a geographical decomposition which produces RMP and subproblems with significantly reduced size/difficulty, so as to improve the efficiency of the solution process. Sub-optimality introduced by the decomposition is cured by an aggregation phase which enable wider combination choices across subgroups.

All in all, most previous literature has modelled the heterogeneous containers delivery problem as a VRP and then developed heuristics to find solution. Instead, in this paper an optimization model is developed based on assignment problem. The aforementioned literatures are extended by considering more realistic restrictions based on a more efficient assignment model structure. Although that this assignment MILP model can combine at most four containers in a single route, it solves much more efficiently than the VRP model, which allows optimal solution to be found for considerably large problems without any heuristics. The decomposition and aggregation approach as proposed thereafter, for practically large instances, relies on the usage of this yet efficient assignment model.

3 Problem Description and Optimization Model

This work deals with the combination of containers on road delivery. An *order* here is defined as the request of transporting a 20ft or 40ft container from a known origin to a known destination. In this sense, both loaded and empty containers can be modelled in the same fashion; the only difference between them is the payload weight. In this study we assume that all orders either start from the port or finish at the port. This is a reasonable assumption for typical loaded import and export orders. For empty container transportation, however, this assumption means we exclude the usage of inland empty storages (depots) and all direct empty transportation between customers from consideration. Nevertheless, the designed model is useful in reality in case when (a) all loaded imports return their container to the port, after delivery at the customer; (b) all loaded exports are serviced by driving an empty container from the port, before collection at the customer. These are standard operational rules implemented by many shipping lines in the market. Note further that for depot turn of empties, one can exclude such empty requests (from customer location to inland depots) at the beginning, use this model to find the optimal solution and then insert the empties into the planned closed-route from and to the port. This will largely simplify the decision process, but give reasonably good solutions since the empty transportation is normally un-time-windowed, and allocated to the closest depot so shorter-distanced than the loaded transportations. The insertion process is not expensive and integrated well with the decomposition.

The optimization model is designed from the perspective of a haulage company, who owns a heterogeneous fleet consisting of both 20ft and 40ft truck heads and chasses, and receives orders from shipping lines well ahead of the delivery day. It is assumed that the whole container is ready for collection from its origin and can be left as a whole at its destination, without worrying about how and when the cargo will be discharged from/loaded into the container. This assumption is consistent with the *strip* strategy used in many literatures of container transportation (see Table 1 and 2), which reflects the practical situation for a haulage company who is only responsible for sending the containers in the desired time window using its own fleet rather than managing the overall import and export plan. The shipping line, on the other hand, is responsible to create the order requests (includes both loaded and empty) according to their business needs, which consists of full information about the origin, destination, time window etc. for every container they would like to move. Under this context, an empty space on the truck is created following the delivery of an order, and all empty containers on the request can be treated in the same way as the loaded ones.

Based on these assumptions, a decision should be made on how to construct the delivery routes to satisfy all demands with the minimum cost, given that a 40ft truck can carry two 20ft containers at a time. The cost considered are travelling costs of the truck which reflects the mileage/fuel cost and the potential rental cost for extra trucks and drivers needed to meet all demands. Figures 1 gives all possible configurations of the routes.

3.1 Parameters

In this Section we summarise the parameters used. Let \mathcal{N} be a set of containers which consists of: \mathcal{P}_1 as a set of 20ft import orders, \mathcal{P}_2 as a set of 20ft export orders, \mathcal{P}_3 as a set of 40ft import orders and \mathcal{P}_4 as a set of 40ft export orders. Each order has an allocated time window $[T_i^s, T_i^e]$ during which the collection/delivery should be made. The payload weight of containers is denoted by $W_i, i \in \mathcal{N}$, which is zero for empty containers. The weight of the truck chassis is denoted by V . Weight limit V_{max}^2 is defined as the total maximum weight a 40ft chassis can carry. Note that following previous discussions, here we represent the needs of taking a container from the port to a customer location as an import order, regardless this container is a real import (loaded) order or a request of re-locating an empty container for the following export services. This means in this study, we ignore the fact that some containers might be loaded and some others empty, and simply describe an order as a customer request of delivering a container from its origin to its destination. We are not concerned about how the empty container will be processed and re-located after fulfilling the current delivery request. If there is such a need, a new order could be generated to capture it.

Every order should be collected from its origin: either from the port (L_o) for import orders or from a customer location ($L_i, i \in \mathcal{N}$) for export orders, after a specific available time (A_i). The handling time is assumed the same for all locations and is denoted by O . Let $f(L_i, L_j)$ and $t(L_i, L_j)$ denotes the travelling cost and time from location L_i to location L_j , respectively. The total working time of drivers, which includes the travelling

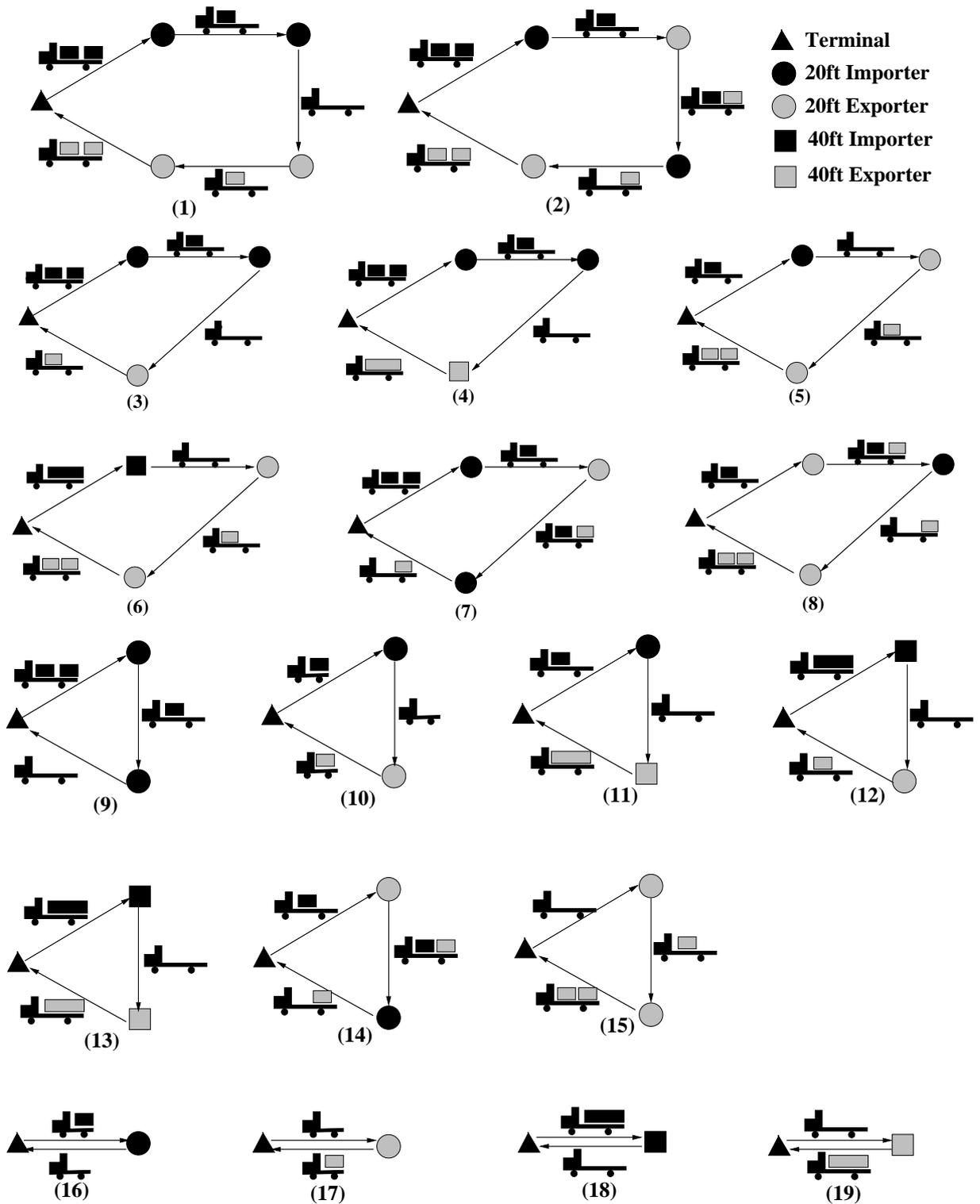


Figure 1: Route configurations

Config. (g)	Route (ξ)	Arcs travelled (\mathcal{A}_ξ)
1	$port \rightarrow i \rightarrow j \rightarrow k \rightarrow l \rightarrow port, \forall i \neq j \in \mathcal{P}_1, \forall k \neq l \in \mathcal{P}_2$	$(L_o, L_i), (L_i, L_j), (L_j, L_k), (L_k, L_l), (L_l, L_o)$
2	$port \rightarrow i \rightarrow k \rightarrow j \rightarrow l \rightarrow port, \forall i \neq j \in \mathcal{P}_1, \forall k \neq l \in \mathcal{P}_2$	$(L_o, L_i), (L_i, L_k), (L_k, L_j), (L_j, L_l), (L_l, L_o)$
3	$port \rightarrow i \rightarrow j \rightarrow k \rightarrow port, \forall i \neq j \in \mathcal{P}_1, \forall k \in \mathcal{P}_2$	$(L_o, L_i), (L_i, L_j), (L_j, L_k), (L_k, L_o)$
4	$port \rightarrow i \rightarrow j \rightarrow k \rightarrow port, \forall i \neq j \in \mathcal{P}_1, \forall k \in \mathcal{P}_4$	$(L_o, L_i), (L_i, L_j), (L_j, L_k), (L_k, L_o)$
5	$port \rightarrow i \rightarrow k \rightarrow l \rightarrow port, \forall i \in \mathcal{P}_1, \forall k \neq l \in \mathcal{P}_2$	$(L_o, L_i), (L_i, L_k), (L_k, L_l), (L_l, L_o)$
6	$port \rightarrow i \rightarrow k \rightarrow l \rightarrow port, \forall i \in \mathcal{P}_3, \forall k \neq l \in \mathcal{P}_2$	$(L_o, L_i), (L_i, L_k), (L_k, L_l), (L_l, L_o)$
7	$port \rightarrow i \rightarrow k \rightarrow j \rightarrow port, \forall i \neq j \in \mathcal{P}_1, \forall k \in \mathcal{P}_2$	$(L_o, L_i), (L_i, L_k), (L_k, L_j), (L_j, L_o)$
8	$port \rightarrow k \rightarrow j \rightarrow l \rightarrow port, \forall j \in \mathcal{P}_1, \forall k \neq l \in \mathcal{P}_2$	$(L_o, L_k), (L_k, L_j), (L_j, L_l), (L_l, L_o)$
9	$port \rightarrow i \rightarrow j \rightarrow port, \forall i \neq j \in \mathcal{P}_1$	$(L_o, L_i), (L_i, L_j), (L_j, L_o)$
10	$port \rightarrow i \rightarrow k \rightarrow port, \forall i \in \mathcal{P}_1, \forall k \in \mathcal{P}_2$	$(L_o, L_i), (L_i, L_k), (L_k, L_o)$
11	$port \rightarrow i \rightarrow k \rightarrow port, \forall i \in \mathcal{P}_1, \forall k \in \mathcal{P}_4$	$(L_o, L_i), (L_i, L_k), (L_k, L_o)$
12	$port \rightarrow i \rightarrow k \rightarrow port, \forall i \in \mathcal{P}_3, \forall k \in \mathcal{P}_2$	$(L_o, L_i), (L_i, L_k), (L_k, L_o)$
13	$port \rightarrow i \rightarrow k \rightarrow port, \forall i \in \mathcal{P}_3, \forall k \in \mathcal{P}_4$	$(L_o, L_i), (L_i, L_k), (L_k, L_o)$
14	$port \rightarrow k \rightarrow j \rightarrow port, \forall k \in \mathcal{P}_2, \forall j \in \mathcal{P}_1$	$(L_o, L_k), (L_k, L_j), (L_j, L_o)$
15	$port \rightarrow k \rightarrow l \rightarrow port, \forall k \in \mathcal{P}_2, \forall l \in \mathcal{P}_2$	$(L_o, L_k), (L_k, L_l), (L_l, L_o)$
16	$port \rightarrow i \rightarrow port, \forall i \in \mathcal{P}_1$	$(L_o, L_i), (L_i, L_o)$
17	$port \rightarrow i \rightarrow port, \forall i \in \mathcal{P}_2$	$(L_o, L_i), (L_i, L_o)$
18	$port \rightarrow i \rightarrow port, \forall i \in \mathcal{P}_3$	$(L_o, L_i), (L_i, L_o)$
19	$port \rightarrow i \rightarrow port, \forall i \in \mathcal{P}_4$	$(L_o, L_i), (L_i, L_o)$

Table 3: Configuration list

time, the handling time and potential waiting time to meet the time window requirements, is restricted by the GB domestic working hour regulation (on-duty hours) for drivers, with the longest working time per day denoted by T_{max} .

Here we ignore the more sophisticated policies such as the total driving time cannot exceed 10 hours (where T_{max} is 11 hours) and the driver has to have a 30-45 minutes break after every 5.5 hours driving. However we argue that the ignored conditions can be met implicitly in most cases with the proposed model, as the handling time for containers is normally two hours which already covers the difference between driving and working, and the break time for drivers.

Let H_1 and H_2 denote the number of 20ft and 40ft trucks owned by the haulage company. In case the available fleet is not enough to send all orders, a per-unit cost $P_i, i = 1, 2$ is paid for every extra truck needed according to the truck type, which can be interpreted as the rental of the truck/chassis and the daily payment to the extra driver.

3.2 Decision Variables

Binary variables x_ξ are created to denote the decision on which route combination should be selected to serve the orders. All possible route configurations with the arcs they are travelling are summarised in Table 3. Continuous variables y_1 and y_2 are defined to capture the additional number of trucks needed to perform all allocated jobs (which could only take integer values in the solution as explained later). In addition, v_i is defined to capture the arrival time of order i at its destination, while s_ξ and e_ξ are defined to capture the starting and finishing time of the route ξ .

Sets

- \mathcal{G} : the set of all configurations as shown in Table 3.2.
- $\mathcal{R}_g, \forall g \in \mathcal{G}$: the set of routes under configuration g .
- $\mathcal{R} := \cup_{g \in \mathcal{G}} \mathcal{R}_g$: the set of all possible routes.
- $\mathcal{R}(i)(\mathcal{R}_g(i))$: the set of routes (under configuration g) serving order i .
- \mathcal{A}_ξ : the set of arcs travelled by route $\xi \in \mathcal{R}$.

Binary Variables

- $x_\xi = \begin{cases} 1, & \text{if route } \xi \text{ is chosen} \\ 0, & \text{otherwise} \end{cases}, \forall \xi \in \mathcal{R}$

Continuous Variables

- $v_i, \forall i \in \mathcal{P}$: arrival time of order i at its customer location.
- $y_i, i = 1, 2$: extra number of trucks/chasses needed other than H1 (20ft) and H2 (40ft).
- $s_\xi, \forall \xi \in \mathcal{R}$: starting time of route ξ from the port.
- $e_\xi, \forall \xi \in \mathcal{R}$: finishing time of route ξ at the port.

3.3 Mathematical Model

A Mixed Integer Linear Programming (MILP) model is formulated to find the best order combinations in transportation, subject to practical restrictions such as weight limits, time windows and working hour regulations.

$$\min \sum_{\xi \in \mathcal{R}} f_{\xi} x_{\xi} + \sum_{i=1,2} P_i y_i \quad (3.1)$$

$$s.t. \sum_{\xi \in \mathcal{R}(i)} x_{\xi} = 1, \forall i \in \mathcal{P} \quad (3.2)$$

$$V + (W_i + W_j) \leq V_{max}^2 + M_1(1 - \sum_{\xi \in \mathcal{R}(i) \cap \mathcal{R}(j)} x_{\xi}), \forall i \neq j \in \mathcal{P}_1 \quad (3.3)$$

$$V + (W_j + W_k) \leq V_{max}^2 + M_1(1 - \sum_{\xi \in \cup_{g \in \{2,7,8,14\}} (\mathcal{R}_g(j) \cap \mathcal{R}_g(k))} x_{\xi}), \forall j \in \mathcal{P}_1, \forall k \in \mathcal{P}_2 \quad (3.4)$$

$$V + (W_k + W_l) \leq V_{max}^2 + M_1(1 - \sum_{\xi \in \mathcal{R}(k) \cap \mathcal{R}(l)} x_{\xi}), \forall k \neq l \in \mathcal{P}_2 \quad (3.5)$$

$$v_j \geq v_i + O + t_{i,j} - M_2(1 - x_{\xi}), \forall \xi \in \mathcal{R}, (L_i, L_j) \in \mathcal{A}_{\xi} \quad (3.6)$$

$$v_i \geq s_{\xi} + O + t_{o,i} - M_2(1 - x_{\xi}), \forall \xi \in \mathcal{R}, (L_o, L_i) \in \mathcal{A}_{\xi} \quad (3.7)$$

$$s_{\xi} \geq A_i + O - M_2(1 - x_{\xi}), \forall i \in \mathcal{P}_1 \cup \mathcal{P}_3, \xi \in \mathcal{R}(i) \quad (3.8)$$

$$e_{\xi} \geq v_i + O + t_{i,o} - M_2(1 - x_{\xi}), \forall \xi \in \mathcal{R}, (L_i, L_o) \in \mathcal{A}_{\xi} \quad (3.9)$$

$$e_{\xi} - s_{\xi} \leq T_{max} + M_2(1 - x_{\xi}), \forall \xi \in \mathcal{R} \quad (3.10)$$

$$T_i^s \leq v_i \leq T_i^e, \forall i \in \mathcal{P} \quad (3.11)$$

$$\sum_{\xi \in \cup_{g \in \{10,16,17\}} \mathcal{R}_g} x_{\xi} \leq H_1 + y_1 \quad (3.12)$$

$$\sum_{\xi \in \cup_{g \in \mathcal{G} \setminus \{10,16,17\}} \mathcal{R}_g} x_{\xi} \leq H_2 + y_2 \quad (3.13)$$

$$x_{\xi} \in \{0, 1\}, \forall \xi \in \mathcal{R} \quad (3.14)$$

$$v_i \geq 0, \forall i \in \mathcal{P} \quad (3.15)$$

$$e_{\xi} \geq 0, s_{\xi} \geq 0, \forall \xi \in \mathcal{R} \quad (3.16)$$

$$y_1, y_2 \geq 0 \quad (3.17)$$

The objective (3.1) is to minimize the total operations cost, which consists of the travelling cost and the penalty for extra trucks needed. Constraint (3.2) force all orders to be delivered, combined or individually. Constraints (3.3) - (3.5) are designed to ensure that the maximum weight allowance is not violated for the combined route on 40ft chasses, which may carry two 20ft containers at the same time. Note that such constraint for individual delivery is left out simply because we assume all orders considered can be feasibly carried by the corresponding type of chassis if it's transported alone, as otherwise we should use alternative methods to deliver it.

The arrival time of all orders to their destinations is calculated by constraints (3.6) - (3.7) and bounded by constraint (3.11) within the desired time window, while constraints (3.8) and (3.9) are used to calculate the starting and finishing time of route at the port. Constraint (3.10) is to ensure that the total time of each route is less than the maximum allowed working time which means that drivers won't work longer than the regulation. Note that we don't have this constraint for single delivery simply because we assume all

orders can be transported on its own within one working shift. Constraints (3.12) and (3.13) calculate how many extra trucks of both types are needed. Finally, constraints (3.14) - (3.18) define the domain of variables. Note that in theory variable $y_i, i = 1, 2$ should belong to non-negative integers ($y_i \in \mathbb{Z}_+ \cup \{0\}$), while in our model it is relaxed to continuous as the objective will force y to take the smallest feasible integer value.

The big M_1 in constraints (3.3) - (3.5) is equal to the maximum of the total weight of any two 20ft containers plus the chassis and taking away the V_{max}^2 , i.e. $M_1 = V + \max\{W_i + W_j | \forall i \neq j \in \mathcal{P}_1 \cup \mathcal{P}_2\} - V_{max}^2$, which represents the upper bound of the l.h.s. of the corresponding constraint. While the big M_2 in constraints (3.6) - (3.10) reflects the maximum value of possible arrival time of every order, which can be defined as the end of the day since the plan is made for one day only. To cover the potential extreme cases when one potential route may take even longer than a day, the M_2 can be defined as the length of time covering two days.

4 Heuristic decomposition and aggregation approach

The container combination problem for inland transportation is classified as NP-hard (Popovic et al, 2014), which means exact optimization approaches can only solve limited size instances. In Section 3 we have formulated a MILP model which solves problems with up to 120 orders in acceptable time by CPLEX (see Section 5.2 for details). In practice, however, a port may operate thousands of containers per day by road. Therefore, an efficient solution approach, e.g., heuristic method, is needed. A well designed heuristic has the capability of finding good solutions for large problems in reasonable time, which in this application is, several hours. In this section a decomposition-and-aggregation heuristic is developed. The method consists of two stages: 1. decompose orders into small subgroups and find a solution for each subgroup by using the formulated MILP model, 2. remove the “well-combined” orders (see Section 4.4 for the definition of “well-combined” orders) and aggregate the remaining to form the second stage MILP model(s). Column generation is applied in both stages (if second stage also contains multiple subgroups) to balance fleet sizes assigned to subgroups.

4.1 Decomposition of orders

Stage 1 aims to reduce the solution time so as to increase the size of the problem that can be managed. A simple decomposition method based on customer locations is therefore proposed. Considering the aim of combining import and export orders to form a closed delivery route starting and finishing at the port, in this study we deploy the fan-shaped clustering approach (Gillett and Miller (1974)) to enable geographical decomposition of customer locations. This approach gives better opportunity of creating closed routes starting and ending at the port (Wren and Holliday (1972)) and therefore is believed to be more suitable than other decompositions strategies such as rectangle-shaped (Daganzo (1984a,b)), ring-shaped (Fang et al. (2013)), seed-based (Fisher and Jaikumar (1984)) decompositions

etc. on this application. However we acknowledge all drawbacks of the approach such as its unsuitability to non-convex service areas.

To create the decomposition, a polar angle of each order's location with respect to the port and the baseline of the seashore is calculated and denoted by $A_i, \forall i \in \mathcal{N}$. Then subgroups are defined by sweeping through the angels from one end to another until a pre-defined number of orders have been covered by each subgroup (see Algorithm 1). As in practice the customer locations are not uniformly distributed, this approach ensures to divide the whole area into subareas with similar number of orders. The number of subgroups we are aiming to have is represented by S , which is an ad-hoc decision balancing between the solution speed and combination opportunities it offers to every subgroup. A brief discussion about how to choose an appropriate S can be found in Section 5.3.

Algorithm 1 - Subgroup definition

Step 0: Let $\bar{n} = \lceil \frac{|\mathcal{N}_i|}{S} \rceil$ denote the expected number of orders per subgroup. Let:

$$\begin{aligned} A_{min} &= \min\{A_i, i \in \mathcal{N}\}; \\ A_{max} &= \max\{A_i, i \in \mathcal{N}\}; \\ s &= 1; \\ d &= A_{min} + 1. \end{aligned}$$

Step 1: *while* $d < A_{max}$, do:

Step 2: Sweep through the angle between A_{min} and d , count how many orders are included and denote by $n^{(s)}$.

Step 3: Check the number of orders against expected number:

if $n^{(s)} \geq \bar{n}$ *or* $d \geq A_{max}$ *then*:

$$\begin{aligned} A^{(s)} &= [A_{min}, d]; \\ A_{min} &= d; \\ s &= s + 1; \end{aligned}$$

and go to **Step 1**.

else: $d = d + 1$ and go to **Step 2**.

Step 4: Return the defined angle range for subgroups: $A^{(s)}, s = 1, \dots, S$.

Obviously the decomposition stops the combination of orders between different subgroups, and introduces a question on how to allocate fleets to serve each subgroup so as to minimize the total delivery cost. It is not hard to imagine, too small fleet sizes may lead to large penalty on extra trucks and too large fleet may lead to capacity loss. Especially in case when capacity is very limited, how to allocate them amongst service areas forms a vital decision. As an initial attempt, we allocate the number of fleets in each subgroup $s = 1, \dots, S$ simply according to the proportional number of the corresponding container

sizes. Suppose $n_1^{(s)}, n_2^{(s)}, n_3^{(s)}, n_4^{(s)}$ represent the number of 20ft import, 20ft export, 40ft import 40ft export orders in groups s , the number of 20ft ($H_1^{(s)}$) and 40ft ($H_2^{(s)}$) trucks are calculated by:

$$H_1^{(s)} = \left\lceil \frac{n_1^{(s)} + n_2^{(s)}}{|\mathcal{P}_1| + |\mathcal{P}_2|} \times H_1 \right\rceil, \quad H_2^{(s)} = \left\lceil \frac{n_3^{(s)} + n_4^{(s)}}{|\mathcal{P}_3| + |\mathcal{P}_4|} \times H_2 \right\rceil \quad (4.1)$$

where $\lceil \cdot \rceil$ represents the function of rounding fractional value to its nearest integer. Then a MILP as presented in Section 3.3 is constructed for each subgroup to find the optimal combination within it.

4.2 Column Generation

In order to balance the fleet size for all subgroups and to decrease the cost gap between the heuristic decomposition and the MILP model, a column generation method is developed. Column generation is an iterative method (Ford Jr and Fulkerson (1958)) which consists of two parts: subproblem(s) to generate column(s) to enter the basis (potential components of optimal solution), and a Restricted Master Problem (RMP) to maintain all generated columns. They are solved iteratively, with the master problem to inform the shadow-price of global constraints and subproblems to identify whether the master problem should be enlarged with additional columns or not. Column generation procedure alternates between the master problem and the subproblem, until the former contains all necessary variables for an optimal solution.

4.2.1 Subproblem

Using Algorithm 1 we can classify all orders into non-overlapping subgroups, which we call decomposition. Decomposition prevents orders in one subgroup to be combined with those in another. Therefore, the subproblem, which decides how orders can be combined for transportation, becomes a separable problem with respect to subgroups. Suppose a subgroup is formed, how an order is delivered (constraint 3.2), the weight limit (constraints 3.3 - 3.5), the departure/arrival time of the order and the working regulation for the combined routes (constraints 3.6 - 3.11) are all irrelevant to other subgroups. Nevertheless, all subgroups request trucks from the same fleet. Although we have presented a formula in Section 4.1 to split trucks into subgroups, it is actually challenging to forecast how many each subgroup would need without attempting to solve the routing problem. It follows that the constraints on fleet size (3.12 - 3.13) become the ‘‘global constraints’’ which prevent the subproblem to be considered separately. Therefore we define the dual variables for the fleet constraints and form a relaxed problem:

$$\min f(x) + \pi_1(H_1 + y_1 - \sum_{\xi \in \cup_{g \in \{10,16,17\}} \mathcal{R}_g} x_\xi) + \pi_2(H_2 + y_2 - \sum_{\xi \in \cup_{g \in \mathcal{G} \setminus \{10,16,17\}} \mathcal{R}_g} x_\xi) \quad (4.2)$$

$$s.t. (3.2) - (3.11), (3.14) - (3.17) \quad (4.3)$$

where $f(x)$ represents the x parts in the original objective (3.1), π_1, π_2 are the dual variables for constraints (4.5) and (4.6) in the RMP, respectively. As said optimization problem (4.2) is decomposable for subgroups, solving which will create for every subgroup an delivery plan satisfying all subproblem constraints. This delivery plan corresponds to a variable in the RMP of minimum reduced cost. However the fleet every subgroup uses, might not be consistent with the total number that is available. This is going to be dealt with in the Restricted Master Problem (RMP).

Every time we obtain a solution of (4.2) for subgroup s , suppose its objective is negative, we define a column (s, f_c^*, q_c^1, q_c^2) whose first element indicates which subproblem it solves, the second gives the optimal delivery cost, the third and the fourth show the number of 20ft and 40ft trucks it uses, respectively. We further assemble all columns for the same subgroup into sets denoted by $\mathcal{C}_s, s = 1, \dots, S$. This information is then passed to the RMP for further iterations. If the solution of all subproblems has non negative objective the iteration is terminated, otherwise the RMP is enlarged with the newly added columns and resolved.

4.2.2 Restricted Master Problem (RMP)

As said, the fleet size restrictions have been left out from the subproblems since they are global constraints which link subgroups together. In the RMP, we retain them and calculate their shadow prices. The original indicator $z_i, i = 1, 2$ for the number of extra trucks used for both types and the corresponding penalty, are also brought into the RMP. Let \mathcal{C}_s denote the set of all columns with negative reduced costs that have been generated from the iterative calling of subproblem s , and $\mathcal{C} = \cup\{\mathcal{C}_s, s = 1, \dots, S\}$. The RMP can be formulated as follows:

$$\min \sum_{c \in \mathcal{C}} f_c^* \lambda_c + \sum_{i=1,2} P_i y_i \quad (4.4)$$

$$s.t. \sum_{c \in \mathcal{C}} q_c^1 \lambda_c \leq H_1 + y_1; \quad (4.5)$$

$$\sum_{c \in \mathcal{C}} q_c^2 \lambda_c \leq H_2 + y_2; \quad (4.6)$$

$$\sum_{c \in \mathcal{C}_s} \lambda_c = 1, \forall s; \quad (4.7)$$

$$\lambda_c \in [0, 1], \forall c \in \mathcal{C}; \quad (4.8)$$

$$y_i \geq 0, i = 1, 2 \quad (4.9)$$

Each column is associated with a variable λ_c , which is restricted between 0 and 1 and indicates the number of times a column is chosen in the solution. The q_c^1 and q_c^2 , represent the number of 20ft and 40ft trucks used by column c . Solving the RMP we obtain the optimal dual variables π_1 and π_2 for fleet constraints, which is needed in subproblems for further iterations. The whole process continues until no new columns with negative

reduced costs are generated. The optimal solution in terms of $\lambda_c, c \in \mathcal{C}$ is then rounded to the nearest integer. Note that in our experiments, although we solve the RMP as ordinary linear program, the optimal λ_c values it generates are very close to 0 or 1 in nearly all cases we've tested. This means that there is always a single column for each subarea that outperforms others in terms of the trade-off between the number of trucks used and the delivery cost. This might be due to the fact that the delivery cost reduction with additional fleets is not so significant comparing to the penalty of violating the fleet constraint. Since the data we've used for the test are taken from real-world applications so in this work we won't go through the more sophisticated Branch-and-Price (B&P) steps to find the integer optimal solution, while we note that B&P should be considered to explore the integer optimal suppose large gaps are observed between the LP objective and the objective value of the rounded solutions. In this work, instead, improvements will be concentrated on the reduction of decomposition gaps rather than the integrality gaps in later sections since decomposition is believed as the main reason for the sub-optimality gaps in this context.

4.3 An example

In this section we show an example to illustrate how the decomposition-column generation approach works. Geographical locations of the port and 20 randomly generated orders are shown in Figure 2. Note that here we use artificial data with uniformly distributed customer locations simply to demonstrate the methodology. More practical tests with real data will be shown in Section 5.

Figure 2: Geographical location of the port and a set of orders

The order set comprises of: $8 \times 20\text{ft}$ import orders, $6 \times 20\text{ft}$ export orders, $3 \times 40\text{ft}$ import orders and $3 \times 40\text{ft}$ export orders. The available fleet sizes are $8 \times 20\text{ft}$ trucks and $6 \times 40\text{ft}$ trucks. The example is firstly solved as a whole by the MILP model (Figure 3). In this case, most orders are delivered jointly with others, created two routes combining four orders, one route combining three orders and four routes combining two orders. Only one order is delivered individually (the order 7). As shown in Table 3, the minimum cost is 1194 and the number of the used trucks are $2 \times 20\text{ft}$ and $6 \times 40\text{ft}$, respectively. We can see when there is no decomposition, the model tends to use as many 40ft trucks as possible to reduce empty trips.

Figure 3: Optimal solution from MILP

Now we decompose the example into four subgroups like what is shown in Figure 4, and solve for each area separately. Comparing to the MILP solution, the decomposition uses relatively more 20ft ($6 \times 20\text{ft}$) trucks and relatively less 40ft ($5 \times 40\text{ft}$), although there are $6 \times 40\text{ft}$ available. Since it creates significantly less four combined routes (obviously

none in this example), the minimum cost obtained from the decomposition is 1505, which is 26% higher than the cost of the real optimal.

Figure 4: Fan-shaped decomposition and the solution

To automatically balance fleet sizes amongst subgroups, column generation approach as stated in Section 4.2 is applied and the result is shown in Figure 5. In this case, one more 40ft truck is allocated to the second subgroup, which enables the optimal route combination of orders 5, 11, 2 and 14 again. While it still prevents the combination of orders 5, 11, 2 and 14 since they are now fall into three different subgroups. This reduces the total cost by 11.43% to 1333, and used the full truck load of four out of six 40ft trucks. It is clear from the result that the cost is still 11.64% higher than the MILP. In next section, we will propose an aggregation step which forms manageable size problems while allows the cross-group order combinations again.

Figure 5: Fan-shaped decomposition with column generation and the solution

# Orders	group index	# Orders in group	# fleet				# Indiv. routes	# 2 com- bined routes	# 3 comb. routes	# 4 comb. routes	Min. cost of groups	Total cost	CPU Time(sec)
			available		used								
			20ft	40ft	20ft	40ft							
MILP - Optimal solution													
20	-	-	8	6	2	6	1	4	1	2	-	1194	3.22
Decomposition solution													
20	1	5	2	1	2	1	2	-	1	-	241	1505	1.76
	2	6	3	1	3	1	2	2	-	-	613		
	3	5	2	2	-	2	-	1	1	-	508		
	4	4	1	2	1	1	-	2	-	-	143		
Decomposition with column generation solution													
20	1	5			2	1	2	-	1	-	241	1333	3.44
	2	6			1	2	2	-	-	1	441		
	3	5	8	6	-	2	-	1	1	-	508		
	4	4			1	1	-	2	-	-	143		

Table 4: Results of the MILP, decomposition and decomposition with column generation for the example

4.4 Aggregation of orders

Looking at the previous example we can see that the decomposition does introduce significant gaps from the optimal solution, since it prevents the combination of orders across subgroups. In this section we will discuss about how to aggregate to create more combined deliveries so as to further reduce cost. Obviously, in the decomposed optimal solution some routes have fully used the truck load already, e.g. the routes combining four 20ft orders. For this type of routes, relaxing the boundary of decomposition may shift part of the route

combination to its neighbour area but would not significantly reduce the delivery cost since orders in the same subgroup is believed closer from each other. Nevertheless, orders that are loosely combined with others or equivalently, orders that are delivered on routes that still have spare capacity for accommodating one/several more orders, should be the ones to be considered further. Therefore, in the second stage, we remove orders on the “well-combine” routes which include (a) $4 \times 20\text{ft}$, (b) $1 \times 40\text{ft} + 2 \times 20\text{ft}$ and (c) $2 \times 40\text{ft}$, then aggregate all the remaining orders and fleets to form an aggregated problem. Depending on the size of the aggregated problem, one can solve it as a whole if it is manageable, or decompose it again suppose it is larger than 120 orders. According to the real data, best aggregation strategy can be selected by experience considering customer density, driving time, travelling speed and solution time for the MILP. The new decomposition must lead to less number of subgroups than before as it considers reduced number of orders, which creates possibilities for the previously separated orders to be combined in the new decomposition. This process can continue until no more “well-combined” routes are generated, or until no improvements on cost is seen. While through our tests using real data, one iteration of the decomposition-aggregation approach has already reduce the sub-optimality gap to 5% on average (see Section 5.2 for details).

Moving back to the example explained in Section 4.3, based on the Figure 5 solution orders 1, 3, 5, 6, 7, 8, 9, 12, 16, 18, 19 and 20 are delivered in well-combined routes. The cost to deliver them is 651 (see Table 4 for more information). This leaves eight orders for the aggregated problem. As eight is a very manageable size here we simply solve the aggregated problem as a whole. Solution is illustrated in Table 4 and Figure 6.

# Orders	# groups	# of groups	# fleet				# Indiv. routes	# 2 comb. routes	# 3 comb. routes	# 4 comb. routes	Min. cost	Cost of Removed orders	Total cost	CPU time(sec.)
			available		used									
			20ft	40ft	20ft	40ft								
Decomposition with column generation solution														
20	1	5			2	1	2	-	1	-	241			
	2	6			1	2	2	-	-	1	441			
	3	5	8	6	-	2	-	1	-	-	508	-	1333	2.44
	4	4			1	1	-	2	-	-	143			
Aggregation solution														
8	1	8	8	2	2	2	-	4	-	-	594	651	1245	4.65

Table 5: Results for the aggregation of the example

The result in Table 4 shows that the total cost of the decomposition-aggregation approach is 1245, which is 6.6% smaller than the decomposition with column generation approach and being very close to the optimal solution value of 1194. Comparing the optimal delivery routes of the MILP and that of the decomposition-aggregation method, many routes are the same for this small example except the routes for orders 2, 5, 7, 11, 14, 15 and 19. In the optimal solution these orders form a 4-combined route of 20ft orders, a 2-combined route of 40-ft orders and a single 20ft route, whereas in the heuristic solution they form a 3-combined and two 2-combined routes. However, in all methods, all available 40ft trucks are used to reduce the total cost.

Figure 6: Aggregated problem for all remaining orders after decomposition

5 Numerical Results

In this section we present the numerical results for the MILP model, the heuristic decomposition with column generation and the decomposition-aggregation approach. Note that all methods are coded in MATLAB R2015b and executed via CPLEX 12.6.1, on a CPU with an Intel(R)Core(TM)i7-4790 processor.

Numerical experiments are carried out on real data obtained from the Port of Felixstowe, which is the largest container port in the UK. A one-day request list, which consists of 1067 orders (213 20ft-import, 232 20ft-export, 329 40ft-import and 293 40ft-export) from/to 346 customer locations are used as the resource dataset from which our experiment samples (in various sizes) are generated. The service area considered in the dataset spread over a pre-defined area that can be covered within 10 hours of driving from the port. The original dataset consists full information about the order origins/destinations, arrival (ready for collection) times, payload weights, assigned time windows and angle information from the port baseline. The explicit location of every customer node is not available due to disclosure concerns, while road distances between each pair of locations are known. From this dataset, we randomly pick up the desired total number of orders according to needs. Therefore the explicit number of 20ft/40ft, import/export, loaded/empty orders are not controlled throughout the experiments. Fleet sizes are decided by formula (5.1) and (5.2) for every randomly generated order list. (5.1) aims to allocate adequate number of 40ft trucks when none of the 40ft containers are jointly deliverable (due to time windows, etc.) with others. Given this number, (5.2) defines the number of 20ft trucks so as to allow an overall 1.5 times of demanded truck spaces by all 20ft containers. Penalty charge for using extra 20ft head/chassis is £600 and for 40ft it's £800.

$$H_2 = \max\{|\mathcal{P}_3|, |\mathcal{P}_4|\}, \quad (5.1)$$

$$H_1 = \max\left\{\left\lceil \frac{|\mathcal{P}_1| + |\mathcal{P}_2| + 2 \times |\mathcal{P}_3| + 2 \times |\mathcal{P}_4| - 4 \times H_2}{2} \times 1.5 \right\rceil, 0 \right\}. \quad (5.2)$$

5.1 Results for the MILP model

As shown in Table 5, the MILP is tested on instances with 10, 30, 50, 100 and 120 orders, ten samples for each group. As all instances are randomly generated, different test composes of different number of orders in each category. The result shows that the model constructs some routes by combining four orders ($2 \times 20\text{ft}$ import orders + $2 \times 20\text{ft}$ export orders) in some instances, especially when the number of orders increases and so is the density. It is also clear from the result that the model always intends to use the full available 40ft fleet rather than the 20ft trucks, due to the fact that doing pairwise delivery of 20ft containers on 40ft trucks reduces the number of empty trips so as to reduce the total delivery cost.

Even sometimes there are still 20ft trucks free from duty, the optimal solution may still choose to hire extra 40ft trucks to do the transportation as it is more cost-saving. It is obvious from the result that the computation time increases with the size of the numerical instances, although it is fast enough for instances as large as 120 orders to be solved within 10 minutes using the MILP model without any simplifications. Comparing to the VRP-SPDTW model considered widely in the literature which tackles with up to 63 orders (Vidović (2012)), the proposed MILP has nearly doubled the problem size on which optimal solutions can be found. For larger examples, e.g. 150 orders, CPLEX runs out of memory.

5.2 Results for heuristic decomposition and aggregation approach

In this section, the results of the decomposition and aggregation approach on small scale examples will be presented. We firstly demonstrate the full solution for three instances with 100 orders in Table 6 to highlight the differences of approaches, and then summarise the experiment with 10 examples per size group for 50, 100 and 120 orders in Table 7.

Results of applying the decomposition (D), the decomposition-column generation (D+CG) and the decomposition-and-aggregation (D+CG+Agg) on three 100 orders examples are summarised in Table 6 and compared with the optimal solution obtained with MILP. In all test examples, the numbers of subgroups are selected to result in, approximately, 30 orders in each subgroup. This number is consistent with the conclusion we make in Section 5.3, where we test through the influences of different decomposition levels. Note that in the decomposition step (D), we use the same proportional heuristic to split fleet into subgroups, as stated in formula (4.1). Note further that the aggregation step in this experiment is solved as a whole as obviously the size of all left-over orders is manageable for 100 orders example.

Looking at the results, the total cost meet our expectation of having $(D) \geq (D+CG) \geq (D+CG+Ag) \geq (MILP)$, while the time of finding a solution with the corresponding method also increases with their accuracy. The column generation step (D+CG) actively reassign fleets amongst subgroups and therefore saves overall penalty for hiring extra trucks in all testing examples. For example, in Instance 100-2, increasing the number of 20ft trucks from 8(9) to 10(11) for subgroup 1(2) reduces the penalty paid for extra 40ft trucks to cover these orders, which results in a cost reduction of 1218. Doing decomposition, even with column generation to balance fleets, may introduce sub-optimality gap as large as 43.3% as seen in instance 100-1. While this gap is largely reduced by doing the additional aggregation step, in all three examples as tested. Now let us run more experiments to see if large cost reductions can always be achieved by aggregation.

Table 7 and Figure 7 show the result of testing all approaches on 10 randomly generated examples for each size category. Average values of absolute cost and the proportional weakness compared to the optimal solution (MILP solution) are presented in Table 7. The best and worst case weakness and the median of each methodology are shown in the box plot Figure 7. Looking at the result we can see, the aggregation step does largely improve the performance of the proposed methodology, which results in an overall 7.92-13.26% weakness than the true optimum, on average. However, the time it takes to perform

# Orders	Number and Type of Containers				# fleet				# Indiv. routes	# 2 combined routes	# 3 combined routes	# 4 combined routes	Total cost	Penalty cost for extra trucks	CPU time(sec.)
	# 20ft Import	# 20ft Export	# 40ft Import	# 40ft Export	available		used								
					20ft	40ft	20ft	40ft							
10	2	5	1	2	4	2	5	2	4	3	-	-	1853	600	0.13
	2	1	5	2	-	5	2	5	4	3	-	-	2725	1200	0.01
	1	3	5	1	-	5	1	5	3	2	1	-	1620	600	0.11
	1	2	2	5	-	5	-	6	3	2	1	-	1953	800	0.01
	1	4	3	2	2	3	2	3	1	3	1	-	868	-	0.01
	4	1	2	3	2	3	2	3	1	3	1	-	1132	-	0.01
	2	2	1	5	-	5	-	6	2	4	-	-	2025	800	0.01
	2	4	1	3	2	3	2	4	3	2	1	-	1883	800	0.07
	3	1	3	3	3	3	3	3	2	4	-	-	1227	-	0.14
	5	1	2	2	2	5	2	5	2	4	3	-	1468	-	0.01
30	5	4	14	7	-	14	1	16	6	10	-	1	5296	2200	0.08
	6	6	9	9	9	9	6	9	-	15	-	-	3375	-	0.07
	6	7	9	8	8	9	6	9	-	15	-	-	3325	-	0.09
	7	9	8	6	9	8	7	8	1	13	1	-	3618	-	0.35
	7	6	6	11	2	11	2	13	4	8	2	1	4913	1600	0.19
	7	8	8	7	10	8	7	8	1	13	1	-	3227	-	0.18
	9	8	6	7	11	7	8	7	1	13	1	-	3272	-	0.13
	7	10	1	12	-	12	-	17	7	7	3	-	7409	4000	0.25
	5	5	10	10	8	10	5	10	-	15	-	-	3320	-	0.03
	3	6	8	13	-	13	-	16	5	8	3	-	5290	2400	0.03
50	11	14	15	10	11	15	11	15	5	18	3	-	6300	-	1:06
	9	15	11	15	12	15	13	16	9	19	1	-	6860	1400	0.31
	12	8	21	9	-	21	-	27	12	8	6	1	10383	4800	0.47
	10	14	10	16	9	16	10	18	12	11	4	1	7456	2200	0.95
	9	15	10	16	9	16	9	19	11	12	5	-	8160	2400	0.75
	11	11	15	13	14	15	11	15	3	22	1	-	5432	-	0.38
	8	10	17	15	11	17	8	17	2	21	2	-	4994	-	0.35
	8	12	18	12	6	18	6	19	5	16	3	1	6025	800	0.28
	8	16	9	17	6	17	6	22	11	13	3	1	9456	4000	0.56
	10	14	13	13	18	13	14	13	4	23	-	-	5854	-	0.11
100	23	25	27	25	33	27	23	27	2	46	2	-	10031	-	280.41
	20	29	19	32	17	32	17	38	23	21	9	2	15013	4800	279.19
	17	21	24	38	8	38	9	44	16	28	8	1	15855	5400	106.01
	16	24	32	28	24	32	20	32	4	48	-	-	10832	-	201.45
	29	14	26	31	25	31	19	31	5	40	5	-	10089	-	102.36
	18	21	25	36	13	36	21	36	18	35	4	-	15670	4800	101.31
	12	27	26	35	16	35	15	41	22	25	8	1	15550	4800	205.49
	14	23	28	35	17	35	17	38	18	29	8	-	12862	2400	102.45
	20	22	28	30	29	30	22	30	6	44	2	-	10462	-	222.45
	15	27	25	33	20	33	19	37	20	29	6	1	13917	3200	212.37
120	15	31	31	43	17	43	31	43	32	38	4	-	22768	8400	423.18
	19	29	42	30	18	42	17	43	9	42	9	-	13492	800	503.17
	19	36	25	40	19	40	20	48	31	23	13	1	19259	7000	419.13
	23	26	32	39	26	39	26	39	14	47	4	-	13672	-	540.53
	24	26	30	40	23	40	22	42	18	36	10	-	14040	1600	446.89
	23	28	28	41	19	41	20	45	24	30	8	3	16209	3800	566.19
	25	27	34	34	39	34	27	34	2	59	-	-	13300	-	587.21
	19	34	29	38	26	38	26	42	26	33	8	1	16078	3200	576.32
	23	28	41	28	19	41	19	43	6	54	2	-	13945	1600	511.14
	15	31	31	43	17	43	31	43	32	38	4	-	22768	8400	357.88

Table 6: Results for testing the MILP model on different scale of instances.

Instance	group index	# orders in group	# fleet				# Indiv. routes	# 2 comb. routes	# 3 comb. routes	# 4 comb. routes	Fuel cost	Penalty of extra trucks	Total cost	(% weakness than MILP)	CPU time (sec.)
			available		used										
			20ft	40ft	20ft	40ft									
100-1															
MILP	-	-	23	33	15	33	-	44	4	-	9967	-	9967	-	216.64
D	1	33	6	12	5	12	1	16	-	-	3662	-			
	2	38	12	10	11	11	7	14	1	-	4637	800	14283	43.3	7.86
	3	29	5	11	2	14	6	7	3	-	2784	2400			
D+CG	1	22			5	12	1	16	-	-	3662				
	2	21	23	33	11	11	7	14	1	-	4637	3200	14283	43.3	14.75
	3	28			2	14	6	7	3	-	2784				
D+CG+Agg	removed	62	-	-	-	29	-	25	4	-	6311	-			
	1	38	23	4	13	5	-	16	2	-	3737	800	10848	8.8	29.17
100-2															
MILP	-	-	25	31	18	31	2	43	4	-	9959	-	9959	-	109.52
D	1	35	6	14	4	15	4	14	1	-	3661	800			
	2	34	10	9	8	9	2	13	2	-	3775	800		21.3	
	3	31	9	8	9	9	7	9	2	-	3043	-	12079		3.96
D+CG	1	35			4	15	4	14	1	-	3661				
	2	34	25	31	10	8	2	16	-	-	4002	-	10861	9.0	
	3	31			11	8	7	12	-	-	3198				7.88
D+CG+Agg	removed	55	-	-	-	27	-	26	1	-	5836	-			
	1	45	25	4	18	4	2	17	3	-	4214	-	10050	0.9	15.86
100-3															
MILP	-	-	10	39	18	39	16	39	2	-	10895	4800	15695	-	248.93
D	1	36	3	16	5	16	6	15	-	-	4738	1200			
	2	36	5	12	7	12	2	17	-	-	4036	1200	17244	9.9	
	3	28	2	11	3	14	8	7	2	-	3070	3000			51.59
D+CG	1	36			5	16	6	15	-	-	4738				
	2	36	10	39	7	12	2	17	-	-	4036	5000	16700	6.4	
	3	28			1	15	7	6	3	-	2926				78.01
D+CG+Agg	removed	61	-	-	-	29	-	26	3	-	6638	-			
	1	39	10	10	16	11	15	12	-	-	5160	4400	16198	3.2	139.35

Table 7: Comparison of full solutions of all approaches on 100 orders instances

the decomposition-and-aggregation approach is 87% less than solving the MILP for larger examples such as those with 120 orders. This advantage will be more significant once the size of the problem prevents it to be tractable directly by CPLEX.

5.3 Results for different decomposition and aggregation levels

In this section we test through different decomposition and aggregation levels. Obviously the larger per group is the better overall result one can expect, and the longer calculation time in general. We are aiming to find the best number of orders per group to balance the computation time (solution difficulty) and the performance. Here we use 200 orders examples to run the experiment since too small problem size will prevent the discussion into larger number of subgroups. In this test we consider the decomposition levels leading to roughly 25 to 70 orders in every subgroup, which means decompose the entire problem into 3 to 8 subgroups. Similarly in the aggregation stage, to avoid very long solution time

# Orders	MILP		# groups	Decomposition			Decomp. + Column Generation			D + CG + Agg.		
	avr. time (sec.)	avr. cost		avr. time (sec.)	avr. cost	avr. % Weak- ness than MILP	avr. time (sec.)	avr. cost	avr. % Weak- ness than MILP	avr. time (sec.)	avr. cost	avr. % Weak- ness than MILP
50	0.34	6081.5	2	0.04	8376.9	40.45	0.07	8141.2	36.76	0.16	6837.1	13.26
100	231.22	11811.8	3	10.14	14313.5	23.47	36.56	13666.4	17.03	66.85	12703.9	7.92
120	591.73	13892.2	3	13.02	17035.4	22.69	42.53	16418.7	18.57	78.10	15616.5	11.98

Table 8: Results of all approaches on instances with 50, 100 and 120 orders

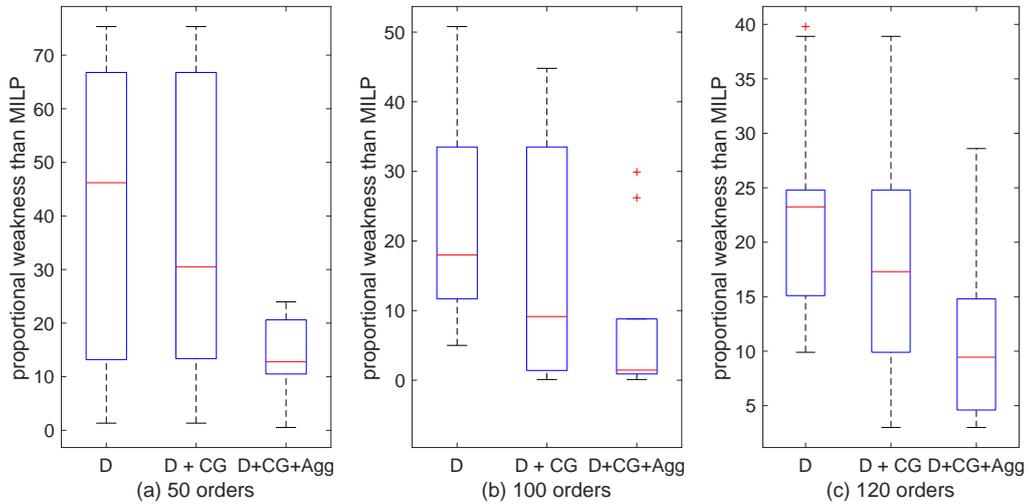


Figure 7: Comparison of all approaches on instances with 50, 100 and 120 orders

we choose the aggregation level that yields up to 70 orders per aggregated group, and solve the resulting MILP to see the influence of different decomposition levels.

Table 8 shows the result of testing different decomposition levels on the same set of 10 randomly generated examples of 200 orders. Since all tests are based on the same set of examples, the absolute costs generated by these methods/decomposition levels are comparable. The decomposition and (D+CG) results meet our expectation of that finer decomposition leads to higher overall costs. Due to this, it is also not hard to understand that finer decomposition in the first phase (decomposition) leaves larger spaces for the aggregation step to work to reduce the final cost. This is why decomposition level 8 receives the highest average proportional improvement by doing aggregation, although with this large improvement the level 8's absolute cost is still higher than simply doing decomposition for 3 levels. On the other hand, the finer decomposition leads to more but smaller MILP problems, which takes much less time than coarser decomposition levels. This is also true through the experiment taking into the fact that with finer decomposition, there might be more orders left after removing well-combined orders from the decomposed solution so as to resulting in larger aggregated problems.

# Orders	# groups	Decomposition		Decomp. + Column Generation			D + CG + Agg.		
		avr. time	avr. cost	avr. time	avr. cost	avr. % im- provement from Decomposition	avr. time	avr. cost	avr. % im- provement from Decomposition
		(sec.)		(sec.)			(sec.)		
200	3	201.36	25993.8	364.38	24690.2	4.96	689.66	24459.4	5.83
	4	47.23	28288.1	106.90	27184	3.90	167.57	25049.1	11.45
	6	6.04	30741.4	22.22	29216.7	4.94	78.29	26110.2	14.80
	8	3.416	35457.8	11.95	33128.6	6.49	67.86	27767.1	21.21

Table 9: Results of all Methods for varies number of groups of 200 orders

As seen in Table 8, decomposing into more subgroups definitely increase total cost, in all three stages of the solution process. Nevertheless, the final (D + CG + Agg) result for the 4-group decomposition is just 2.4% (6.7%) higher than that is for the 3-group, while the latter takes four (nine) times longer to achieve. This suggests that, suppose the solution time is not a matter, decompose the problem into subproblems with the largest manageable size is the best option. However in large-size, real-world applications where resources are limited, decomposing into subgroups with 30-50 orders achieves a good balance between the solution time and optimality. In addition to this, one more interesting point to make is that the (D+CG+Agg) result for 4-levels is better than doing decomposition only for 3 levels (which is what industry normally do in practice), although it takes shorter time. This means the proposed decomposition-and-aggregation process does help in achieving better solutions with less effort, if appropriate decomposition levels are deployed.

5.4 Results for larger instances

Finally, large size instances with 150, 200, 300, 400 and 500 orders are tested. Table 9 summarises the averaged result of solving the large size instances by the three processes on 10 random examples per size category. Note that as optimal solution is not achievable for these large scale problems, results are compared with the industrial-standard decomposition approach only. For each size category, orders are decomposed into different number of subgroups based on the expected number of orders in each subgroup (around 30 to maintain reasonable solution time). Same expected number of orders in each subgroup is used to perform both the decomposition step and the decomposition of aggregated problems with remaining orders. We can see from the result that the average cost saving of the decomposition-column generation is around 5%, while the average improvement after doing the aggregation is over 20%. Both figures are not decreasing as problem size gets larger. Examples with up to 500 orders can be managed efficiently in 45 minutes, on average. It can be seen that the solution time increases linearly with problem size for the Decomposition-only approach. Therefore if the solution time is very restricted, one can also divide the orders together with the fleets completely into parts, and solve each part using the Decomposition-and-aggregation approach independently one from another.

# Orders	# groups	Decomposition		Decomp. + Column Generation			Decomp. + Column Gen. + Aggregation		
		avr. time	avr. cost	avr. time	avr. cost	avr. % im- provement from Decomposition	avr. time	avr. cost	avr. % im- provement from Decomposition
		(sec.)		(sec.)			(sec.)		
150	5	1.68	24774.2	16.34	23578.0	4.72	64.15	20880.5	15.54
200	6	6.04	30741.4	22.22	29216.7	4.94	78.29	26110.2	14.80
300	10	8.05	55368.4	347.81	52020.2	5.32	416.50	44128.2	20.13
400	12	19.80	78169.7	241.28	73333.5	6.27	330.88	60141.3	22.59
500	14	26.70	96512.1	1738.20	91009.4	5.71	2597.09	74996.1	22.25

Table 10: Results for larger instances

6 Conclusions

This paper investigates the transportation of two types of orders: 20ft and 40ft containers on road with both 20ft and 40ft long trucks. For this purpose a Mixed Integer Programming (MILP) model is proposed. The aim of this model is to minimize the transportation cost and the cost of hiring extra trucks when existing fleet is not adequate to cover all demands. Important practical restrictions are captured in this model such as the ready-for-collection time of containers at the port/customer locations, the regulation of the working time and the time windows at customer locations. The MILP model can be implemented to obtain the decision of how to combine orders to form closed delivery routes starting and ending at the port terminal, with up to 4 containers per delivery route. The MILP model is tested for different size of instances drawn from real data in which the result shows that the model is capable to solve efficiently the problem with up to 120 orders by using the CPLEX software package.

In order to deal with larger instances, a decomposition and aggregation heuristic approach is designed in which the locations of orders are decomposed geographically into small subgroups and solved by the formulated MILP model. In order to balance the fleet size amongst all subgroups, column generation method is used. Orders are then aggregated and problem size is reduced by removing the best combined orders. The aggregated problem can then be solved as a whole by constructing a new MILP or decomposed again using the same strategy. Using the proposed methodology, problems with up to 500 orders can be solved within 1 hour. Testing results on real data shows that the proposed decomposition-and-aggregation approach introduces a sub-optimality gap between 5.14-14.61%, but scales well with the size of the problem. For large instances where optimal solution is not achievable, results are compared with an industrial-standard simple decomposition approach. The proposed method improves this decomposition cost by over 20% on average, which is also easily merge-able with the simple decomposition to meet practical needs on the solution time and accuracy.

References

- Braekers, K., Janssens, G., and Caris, A. (2011). Challenges in managing empty container movements at multiple planning levels. *Transport Reviews*, 31(6):681–708.
- Chung, K. H., Ko, C. S., Shin, J. Y., Hwang, H., and Kim, K. H. (2007). Development of mathematical models for the container road transportation in korean trucking industries. *Computers & Industrial Engineering*, 53(2):252–262.
- Chung, L., Pesenti, R., and Ukovich, W. (2006). Minimizing fleet operating costs for a container transportation company. *European Journal of Operational Research*, 171(3):776–786.
- Chung, R. K., Shi, N., Powell, W. B., and Simao, H. P. (2008). An attribute-decision model for cross-border drayage problem. *Transportation Research Part E: Logistics and Transportation Review*, 44(2):217–234.
- Crainic, T. and Kim, K. (2006). Intermodal transportation. *Transportation*, 14:467–537.
- Daganzo, C. (1984a). The distance traveled to visit n points with a maximum of c stops per vehicle: An analytic model and an application. *Transportation Science*, 18(4):331–350.
- Daganzo, C. (1984b). The length of tours in zones of different shapes. *Transportation Research Part B: Methodological*, 18(2):135–145.
- Fang, Z., Tu, W., Li, Q., Shaw, S., Chen, S., and Chen, B. (2013). A voronoi neighborhood-based search heuristic for distance/capacity constrained very large vehicle routing problems. *International Journal of Geographical Information Science*, 27(4):741–764.
- Fisher, M. and Jaikumar, R., . (1984). A generalized assignment heuristic for vehicle routing. *Networks*, 11(2):109–124.
- Ford Jr, L. and Fulkerson, D. (1958). A suggested computation for maximal multi commodity network flows. *Management Science*, 5(1):97–101.
- Funke, J. and Kopfer, H. (2015). A neighborhood search for a multi-size container transportation problem. *IFAC-PapersOnLine*, 48(3):2041–2046.
- Funke, J. and Kopfer, H. (2016). A model for a multi-size inland container transportation problem. *Transp Res Part E Logist Transp Rev*, 89:70–85.
- Gillett, B. and Miller, L. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations research*, 22(2):340–349.
- Gronalt, M., Hartl, R., and Reimann, M. (2003). New savings based algorithms for time constrained pickup and delivery of full truckloads. *European Journal of Operational Research*, 151(3):520–535.

- Hajem, A. D., Yang, X., and Warnes, M. K. (2017). An efficient mixed integer programming model for pairing containers in inland transportation based on the assignment of orders. *Journal of the Operational Research Society*, 68(6):678–694.
- Konings, R. (2005). Foldable containers to reduce the costs of empty transport? a cost–benefit analysis from a chain and multi-actor perspective. *Maritime Economics and Logistics*, 7(3):223–249.
- Lai, M. (2013). Models and algorithms for the empty container repositioning and its integration with routing problems. *PhD thesis, University of Cagliari*.
- Lai, M., Crainic, T., Di Francesco, M., and 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.
- Lübbecke, M. and Zimmermann, U. (2003). Engine routing and scheduling at industrial in-plant railroads. *Transportation Science*, 37:183–197.
- Lun, Y.V., L. K. and Cheng, T. (2010). Shipping and logistics management. *Springer*.
- Morlok, E. and Spasovic, L. (1994). Redesigning rail-truck intermodal drayage operations for enhanced service and cost performance. *Journal of the Transportation Research Forum*, 34(1):16–31.
- Namboothiri, R. and Erera, A. (2008). Planning local container drayage operations given a port access appointment system. *Transportation Research Part E: Logistics and Transportation Review*, 44:185–202.
- Nordsieck, N., B. T. and Schönberger, J. (2016 (online)). Potential of improving truck-based drayage operations of marine terminals through street turns. *Dynamics in Logistics*, pages 433–443.
- Notteboom, T. E. and Rodrigue, J.-P. (2005). Port regionalization: towards a new phase in port development. *Maritime Policy & Management*, 32(3):297–313.
- Parragh, S., K.F., D., and R.F., H. (2008a). A survey on pickup and delivery problems part i: Transportation between customers and depot. *Journal für Betriebswirtschaft*, 58(1):21–51.
- Parragh, S., K.F., D., and R.F., H. (2008b). A survey on pickup and delivery problems part ii: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*, 58:81–117.
- Popović, D., Vidović, M., and Nikolić, M. (2014). The variable neighborhood search heuristic for the containers drayage problem with time windows. *Soft Computing in Industrial Applications*, pages 351–364.

- Reinhardt, L. B., Pisinger, D., Spoorendonk, S., and M., S. M. (2016). Optimization of the drayage problem using exact methods. *INFOR: Information Systems and Operational Research*, 54(1):33–51.
- Reinhardt, L.B., S. S. and Pisinger, D. (2012). Solving vehicle routing with full container load and time windows. *ICCL 2012: Computational Logistics*, pages 120–128.
- Shiri, S. and Huynh, N. (2016). Optimization of drayage operations with time-window constraints. *International Journal of Production Economics*, 176:7–20.
- Shiri, S. and Huynh, N. (2018). Assessment of us chassis supply models on drayage productivity and air emissions. *Transportation Research Part D: Transport and Environment*, 61:174–203.
- Smilowitz, K. (2006). Multi-resource routing with flexible tasks: an application in drayage operations. *Iie Transactions*, 38(7):577–590.
- Stahlbock, R. and VoB, S. (2008). Operations research at container terminals: a literature update. *OR spectrum*, 30(1):1–52.
- UNCTAD (2015). Review of maritime transport. *United Nations publication, New York and Geneva*, Available at <http://unctad.org/en/PublicationsLibrary/rmt2015>.
- Vidović, M., Radivojević, G., and Raković, B. (2011). Vehicle routing in containers pickup up and delivery processes. *Procedia-Social and Behavioral Sciences*, 20:335–343.
- Vidović, M., N. M. . P. D. (2012). Two mathematical formulations for the containers drayage problem with time windows. *International Journal of Business Science and Applied Management*, 7(3):23–32.
- Wang, X. and Regan, C. A. (2002). Local truckload pickup and delivery with hard time window constraints. *Transportation Research Part B: Methodological*, 36:97–112.
- Wen, S. and Zhou, P. (2007). A container vehicle routing model with variable traveling time. *Automation and Logistics, 2007 IEEE International Conference*, pages 2243–2247.
- Wren, A. and Holliday, A. (1972). Computer scheduling of vehicles from one or more depots to a number of delivery points. *Journal of the Operational Research Society*, 23(3):333–344.
- Zhang, R., Yun, W. Y., and Kopfer, H. (2015). Multi-size container transportation by truck: modeling and optimization. *Flexible Services and Manufacturing Journal*, 27(2-3):403–430.