

# Tighter MIP Models for Barge Container Ship Routing

Laurent Alfandari<sup>a</sup>, Tatjana Davidović<sup>b</sup>, Fabio Furini<sup>c</sup>, Ivana Ljubić<sup>a,\*</sup>, Vladislav Maraš<sup>d</sup>, Sébastien Martin<sup>e</sup>

<sup>a</sup>ESSEC Business School, Cergy-Pontoise, France

<sup>b</sup>Mathematical Institute of the Serbian Academy of Sciences and Arts, Belgrade, Serbia

<sup>c</sup>PSL, Université Paris Dauphine, Paris, France

<sup>d</sup>University of Belgrade, Faculty of Transport and Traffic Engineering, Belgrade, Serbia

<sup>e</sup>LCOMS, Université de Lorraine, Metz, France

---

## Abstract

This paper addresses the problem of optimal planning of a line for a barge container shipping company. Given estimated weekly splittable demands between pairs of ports and bounds for the turnaround time, our goal is to determine the subset of ports to be called and the amount of containers to be shipped between each pair of ports, so as to maximize the profit of the shipping company. In order to save possible leasing or storage costs of empty containers at the respective ports, our approach takes into account the repositioning of empty containers. In our setting we determine a single route following the outbound-inbound principle (i.e., the predefined ordering of ports is given, together with the starting and the ending port).

We first propose two new MIP formulations that are tailored for barge container ship routing in the inland waterway transport. We then demonstrate that the models can be extended to general maritime shipping given the outbound-inbound principle. On the publicly available set of benchmark instances for barge container routing, our models significantly outperform the existing approaches from the literature. We also propose some variants of the problem that are of interest for practitioners in the domain, including optimization of the turnaround time, allowing multiple round-trips, and dealing with unsplittable demands. Numerical experiments are provided to compare the computational performance of the models and the impact of both empty container repositioning and unsplittable demands on the total profit.

*Keywords:* Integer Linear Programming, Sea and Inland Waterway Transport, Liner Shipping Network Design, Empty Container Repositioning, Barge Container Ship Routing

---

## 1. Introduction

Liner shipping network design is a family of important and challenging problems in sea and inland waterway transport dealing with a creation of a (set of) sailing route(s) for a designated fleet to transport

---

\*Corresponding author

*Email addresses:* [alfandari@essec.edu](mailto:alfandari@essec.edu) (Laurent Alfandari), [tanjad@mi.sanu.ac.rs](mailto:tanjad@mi.sanu.ac.rs) (Tatjana Davidović), [fabio.furini@dauphine.fr](mailto:fabio.furini@dauphine.fr) (Fabio Furini), [ljubic@essec.edu](mailto:ljubic@essec.edu) (Ivana Ljubić), [v.maras@sf.bg.ac.rs](mailto:v.maras@sf.bg.ac.rs) (Vladislav Maraš), [sebastien.martin@univ-lorraine.fr](mailto:sebastien.martin@univ-lorraine.fr) (Sébastien Martin)

multiple commodities. During the last decades many variants of the liner shipping network design have been addressed in the literature (see, e.g. recent surveys given in [22, 6, 4, 12]). In general, liner shipping companies have to design *lines*, i.e., sequences of calling ports with a given schedule that are operated periodically. In this article we consider the tactical part of this decision making process in which a route for a given liner container ship has to be defined under the following Assumptions:

1. A predetermined ordering of ports for the outbound-inbound trips is given. This is the natural way of scheduling routes in the inland waterway transport.
2. The port calling sequence must start at and return to the first port (in case of barge transport, it is a sea port, located at a river mouth, see Figure 1).
3. The liner ship must stop at the last port (in barge transport, it is the furthest port upstream) where it changes its direction.
4. A single route is planned, hence there is no transshipment.
5. Repositioning of empty containers between the ports is allowed.

For a given liner ship, the problem consists of selecting a subset of calling ports upstream and downstream and, given weekly (splittable) demands of containers between all pairs of ports, deciding what amount of that demand will be shipped in order to maximize total profit within the given planning horizon. In addition to revenues associated with demand units (i.e., containers) between pairs of ports, one has to consider operational costs that include fuel cost, port dues and cargo handling cost. Moreover, the ship must complete its route within a given time interval.

Liner shipping network design under Assumptions 1-4 has been introduced in the seminal paper by [26]. Since then, these concepts have been extended by introducing new and important aspects relevant in the maritime or inland waterway shipping (see Section 2 for the detailed literature overview). However, what remained insufficiently studied in the literature is the important and challenging question studied in the present article: how to develop an integrated approach to design shipping routes while taking into account *empty container balancing and repositioning* at the same time?

According to [34], the containerized cargo flows on major container trade routes are characterized by huge imbalances between inbound and outbound directions, see also Song and Dong [32]. In addition, since the flow of containers has to be balanced at each port, these imbalances result in empty container leasing or storing at respective ports. Shipping the empty containers between the ports instead, has a strong impact on cost calculation. Hence, when determining the liner shipping routes, in some cases the profit of a shipping company can be significantly improved if empty container flows are treated adequately and if their flow is planned simultaneously with the design of the shipping routes. Due to the global financial crisis and the turmoil in global sea freight, the container shipping business is hardly profitable (see [10]). For example, Hanjin, which was the world's seventh largest container shipper, went bankrupt in August 2016.

It is therefore clear that creating profitable lines becomes a key competitive advantage in container shipping business.

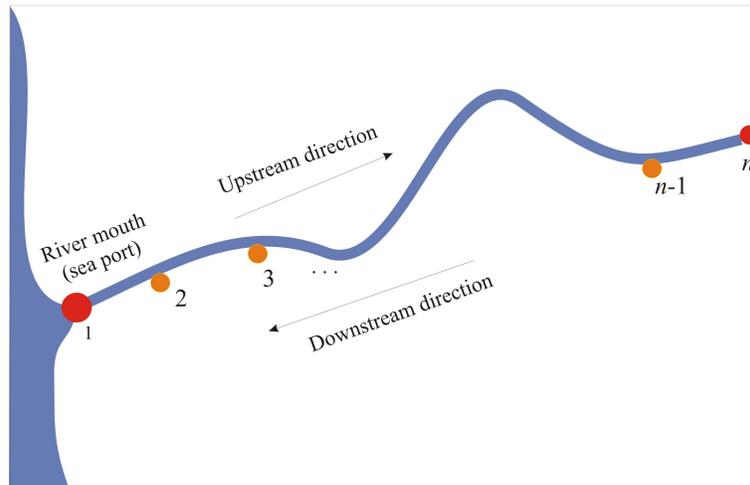


Figure 1: An example of liner shipping along a river with  $n$  ports.

As mentioned above, the basic problem of routing a single container ship while maximizing profit under a knapsack-type time constraint has been studied in [26]. The major assumption in this setting is that the ordering of ports for a given ship is predetermined. To our knowledge, an integrated approach to design the optimal ship route involving empty container repositioning was considered for the first time in [29]. In that article, the authors assume that a pre-ordered list of ports is given and that all container demand emanating from a port must be satisfied if that port is called. However, in their model, a ship can change direction multiple times (at some intermediate ports of call) before returning to the initial port. A problem variant for barge container shipping with outbound-inbound principle and empty container repositioning has been studied in [19].

*Contributions:* In this article, we propose two new MIP formulations that explicitly take advantage of the outbound-inbound principle. This setting is particularly important for barge container shipping, but it is also relevant in maritime routing when the precedence relations between ports are given (e.g., when shipping containers along a coast). In contrast to the existing models from the literature (that require arc-variables for modeling the routes), both new models utilize node-variables for the route design. The first formulation requires arc-variables for modeling empty containers. The second formulation is more compact as it utilizes node-variables only and handles empty containers as a single commodity. An equivalence of the two models, concerning the strength of their LP relaxations, is shown. The two models first apply to the case where the distance traversed between the starting and last port remain constant whatever the port calling sequence, as this is the case for barge container shipping along a river. We then extend our models to the general case

of [26] where shortcuts between ports are allowed (and the total traversed distance can be shortened if some of the ports are not called).

We furthermore show that the problem remains strongly NP-hard, even after relaxing many of its constraints, and we also discuss a special polynomially solvable case.

Finally, we show how to extend the new MIP models so as to (1) optimize the size of the fleet and maximize the profit simultaneously, (2) find the optimal number of round-trips within the planning horizon, or (3) deal with unsplittable demands. With the extension (2) we are addressing the same problem as the one described in [26], while additionally taking the empty container balancing and repositioning into account.

Our computational study is conducted on a set of benchmark instances of barge container shipping from the literature. Our new modeling approach based on node-variables for route design enables us to significantly reduce the computational time and to solve to optimality all instances with up to 25 ports in a few seconds only, thus drastically outperforming the previous state-of-the-art model. For the more challenging variant with unsplittable demands, our approach is able to compute (near) optimal solutions within a short computing time.

The paper is structured as follows. Section 2 gives a detailed overview of the related literature. We then focus on the Barge Container Shipping Problem (BCSP) and in Section 3 we provide the formal problem definition and the NP-hardness proof. Section 4 provides the two new MIP formulations, together with the proof of equivalence between the two models. We then show in Section 5 how to adapt our models for maritime shipping (while assuming the ordering of ports to be given). In this section we study the various extensions mentioned above. Computational experiments on benchmark instances are given and analyzed in Section 6, whereas Section 7 concludes the paper.

## 2. Related Work

A classification of optimization problems for liner container ship routing was given in the recent surveys of [6] and [22]. Following their classification, our problem falls into the category *Liner Container Shipping Network Design* (single route or several routes without transshipment). These two surveys, along with a recent paper [4], cite a dozen of papers published in the last decade in that specific category. An older survey by [5] provides a list of papers on general ship routing and scheduling.

The recent article by [4] provides an excellent overview of the major logistics aspects and challenges for the Operations Research community in the liner shipping business. The authors present a rich integer programming model based on services that constitute the fixed schedule of a liner shipping company (multi-route multi-vessel case). In addition, a publicly available benchmark suite of data instances is created. Unfortunately, the model provided by [4] does not take empty container repositioning into account, and, consequently, their benchmark suite does not contain cargo handling cost associated to empty containers,

nor assumes that the pre-ordering of ports that could be called is given.

*General maritime route design with outbound-inbound principle and without transshipment.* The previously cited paper of [26] falls into this category in which no transshipment is allowed, i.e., exchanging containers between two ships is not an option. In [27] the authors extend their previous model to designing multiple ship routes for a heterogeneous fleet. In both papers it is assumed that the order of ports that could be called is predetermined, with a fixed starting and ending port. In [26] a MIP formulation has been proposed for simultaneously optimizing the total profit and the number of round trips of the ship in a week, the latter being represented by a decision variable  $\alpha$ . Although this leads to a quadratic model, the variable  $\alpha$  can only take a few integer values, so that the authors propose to solve the problem by enumerating all possible values of  $\alpha$ . This boils down to solving the same model for each  $\alpha$  but with a different constraint concerning the total allowed time per route. The authors apply Benders decomposition technique, whereas in the multi-vessel extension of [27] Lagrangian relaxation and decomposition is involved.

*Liner shipping network design with empty container repositioning.*

To our knowledge, the route design with empty container repositioning is considered for the first time in [29]. In this problem variant, pairwise demands are given and profit is to be maximized. In addition, all the cargo traffic between two ports must be satisfied if the ports are called and the ship can change its direction multiple times (at some intermediate ports) before returning to the initial port, which differs from [26]. The authors propose a genetic algorithm to find heuristic solutions. This algorithm explores possible calling sequences of ports, and solves an LP for each given port sequence found during the search, involving empty container variables between two ports. When going in the outbound (or inbound) direction, the authors bring the argument that the ship is allowed to change its direction and move backward to an earlier port, for a matter of empty container repositioning. Such flexibility may indeed provide a more economical solution, but, to our knowledge, it is less accepted by the shipping companies, which is why in our article we keep the assumption that the strict outbound-inbound principle has to be respected.

Table 1 provides a classification of papers on liner shipping route design with empty container repositioning that have been published since the work of [29]. The column “ports selection” refers to papers in which the selection of ports is part of the routing problem. Note that in some of these articles, the calling ports are already given (see the column “pre-specified line services”), and the major decisions concern the shipping of commodities and empty containers. The “inbound-outbound” column indicates the papers that assume inbound-outbound routing, and the “single/multi” column states whether the model deals with the design of a single route, or multiple routes for a fleet of ships.

In [9], a multi-route multi-vessel problem is considered. In [21], the authors design a hub-and-spoke network with multiple routes. In [3], the routes are given, and the problem consists of determining the

amount of containers to be shipped along each route (multi-flow in a network is solved by Dantzig-Wolfe decomposition). In [30], the authors consider a problem with multiple liner ships where demands between ports are to be satisfied while minimizing costs, including transshipment costs, and empty container repositioning and inventory costs. The time dimension is taken into account. There is no pre-specified ordering of ports in the route design. Multiple cargo routes are designed in a first-stage by shortest-path computations inside a MIP, whereas the empty container repositioning is performed in a second-stage. In [31], a single long-haul route is considered for liner shipping, composed of several cycles with pre-specified ordering of ports for each cycle. The relationships between the container flow pattern and the route structure are exploited to simplify the design of the route in the first-stage and to better reposition the empty containers at the second stage. Sizing the fleet of ships assigned to the route and their capacity is performed in a third stage. Multi-route planning with cost minimization is also studied in [11].

The remaining papers from Table 1 deal with barge and inland waterway liner transportation and will be addressed in the following paragraph.

In all papers cited in Table 1, arc variables (associated with pairs of ports) are used to design the ship route and measure the total trip duration that should not exceed the given time limit. That may appear natural when modeling maritime routes since making a shortcut between two ports by skipping an intermediate port could shorten the length of the route, depending on the location of the ports. However, let us note that when routing a barge container ship along a river, skipping a port along the route does not shorten the distance, hence in this particular setting there is no direct justification for using arc variables to design the ship route.

*Liner shipping network design in the inland waterway transport.* We now review papers specifically dealing with inland waterway shipping, since our generic model is particularly suited for routing a barge container ship along a river. As in [26], all these papers deal with selecting the calling sequence of ports for a single line that should respect a predetermined order, both in the outbound and inbound direction, while maximizing profit and respecting a given time limit. The major difference to [26] is that the location of ports along a river induces a fixed travel time between the starting and last port. In addition, as in [29], the balancing and repositioning of empty containers is considered. In [18], the author proposes a MILP formulation with binary variables associated with each pair of ports. This formulation, along with a MILP-based heuristic is implemented in [19] where the authors managed to solve instances with up to 20 ports to provable optimality, but, typically, more than a day of computing was required to provide the optimal solutions. Other papers specifically dealing with barge route design and inland waterway liner transportation with empty container repositioning are [2], [35] and [1]. The three later articles deal with multi-route multi-vessel optimization. The first one considers the selection of unsplitable demands that maximizes profit, whether the other two

Table 1: Classification of route design problems with empty container repositioning

Paper	Empty cont. repositioning	ports selection	pre-specified line services	inbound / outbound	single / multi
[29]	×	×		×	single
[19]	×	×		×	single
[2]	×	×		×	multi
[1]	×	×		×	multi
[9]	×		×	×	multi
[21]	×		×		multi
[3]	×		×		multi
[30]	×		×		multi
[31]	×	×		×	single <sup>e</sup>
[11]	×		×		multi

*a* : possibility of ship turning back; *b* : first and last ports not pre-specified; *c* : network can be slightly more complex than a line; *d* : pre-specified set of potential routes ; *e* : multiple ships are assigned to a single route made of several cycles. Inbound-outbound principle holds for each cycle.

deal with covering demands at minimum cost. Note that all papers used arc variables, both for the route design and the empty container repositioning, which does not exploit the line structure of the route on a river.

### 3. Notation and Problem Definition

In this section we introduce the input parameters, provide a formal problem definition and discuss the problem computational complexity.

The following input is given (where the units of measure used are hours [h], tons [t], twenty-foot equivalent unit [TUE], kiloWatt [kW], kiloWatt hour [kWh] and US dollar [US\$]):

- $N = \{1, \dots, n\}$  : ordered set of  $n$  ports, where 1 is the starting port and  $n$  is the ending port in the outbound direction; the ship should stop at port  $n$  and go back to port 1.
- $D_{ij} \in \mathbb{Z}_+$  : weekly expected number of full containers available to be transported between ports  $i$  and  $j$  [TEU/week];
- $C \in \mathbb{Z}_+$  : capacity of the ship [TEU]
- $P_{ij} \in \mathbb{R}_+$  : freight rate per container from port  $i$  to port  $j$ ,  $i, j \in N$  [US\$/TEU]

- $F_i$  : entry cost per call at port  $i$ ,  $i \in N$  [US\$]
- $L_i^f$  ( $U_i^f$ ): loading (unloading) cost per full container at port  $i$ ,  $i \in N$  [US\$/TEU]
- $L_i^e$  ( $U_i^e$ ): loading (unloading) cost per empty container at port  $i$ ,  $i \in N$  [US\$/TEU]
- $L_i$  ( $S_i$ ): short-term leasing (storage) cost per empty container at port  $i$ ,  $i \in N$  [US\$/TEU]
- $T_i^l$  ( $T_i^u$ ): average loading (unloading) time per full container at port  $i$ ,  $i \in N$  [h/TEU]
- $\tilde{T}_i^l$  ( $\tilde{T}_i^u$ ): average loading (unloading) time per empty container at port  $i$ ,  $i \in N$  [h/TEU]
- $T_i^a$  ( $T_i^d$ ): stand-by time for arrival (departure) at port  $i$ ,  $i \in N$  [h]
- $T_{\min}$ : minimum allowed travel time [h]
- $T_{\max}$ : maximum allowed travel time [h]
- $T_0$  : total sailing time to go from port 1 to port  $n$ , and to go back to port 1 [h]
- $T_w$  : total waiting time for crossing borders and locks on both directions ( $T_w$  is counted in  $T_0$ ) [h].

Note that the sailing time  $T_0$ , a constant in the inland waterway shipping, does not comprise the stopping times at ports for loading and unloading containers. The total travel time is the sum of the sailing time and the stopping time at ports, hence  $T_{max} \geq T_0$ .

Additional parameters are:

- **dcc** : daily time charter cost of a ship [US\$/day];
- **P<sub>out</sub>** : engine output (propulsion) [kW]; ship speed [km/h];
- **fp** (**lp**) : fuel (lubricant) price [US\$/t];
- **scf** (**sc1**) : specific fuel (lubricant) consumption [t/kWh]; %item **t<sub>b</sub>**: total time of border crossings at all borders between ports 1 and  $n$  [h];

The case in which we allow to partially satisfy the demand  $D_{ij}$  between ports  $i$  and  $j$ , is called *splittable demands* in the following, while the case in which either zero or all  $D_{ij}$  containers have to be shipped is called *unsplittable demands*. In this article, both problem variants are addressed.

### 3.1. Formal Problem Definition

In the following, we first address the problem of the barge container liner shipping route design along a river, as it was defined in [19]. Later, in Section 5 we show how to extend this problem into a more general setting.

**Definition 1 (Barge Container Shipping Problem (BCSP)).** *Given the input parameters described above, the BSCP asks to determine the sequence of calling ports, both in the outbound and inbound direction, and the number of full and empty containers to be shipped between the ports, so as to maximize the profit, which is defined as the difference between the revenue for shipping full containers, and the port call cost, cargo-handling cost, and bunker and capital costs, see (1). Thereby, the following constraints need to be respected:*

- *The route must start at port 1 and must visit port  $n$ . Total turnaround time (including traveling and service time, see (3)) must be between  $T_{\min}$  and  $T_{\max}$ ,*
- *At each port  $i$ , if the total inflow of full and empty containers (counting the flow both in the outbound and inbound direction) is not equal to the total outflow, the difference should be balanced by either leasing or storing containers at that port (the balancing of empty containers is explained in the next section). Alternatively, to save the latter cost, empty containers can be transported on the ship.*
- *For practical reasons, in barge container shipping companies prefer solutions in which containers can be transported either in the outbound or in the inbound direction (see [7, 19]). This means that no mixing is allowed, i.e., no container can be loaded in the outbound and unloaded in the inbound direction.*

To define the profit function, let  $N_{out} \subseteq N$  be the set of ports called in the outbound and  $N_{in} \subseteq N \setminus \{n\}$  the subset of ports called in the inbound direction and let  $N' = N_{out} \cup N_{in}$ . For any pair of distinct ports  $i, j \in N_{out}$  or  $i, j \in N_{in}$ , let  $a_{ij}$  and  $b_{ij}$  denote the amount of full, respectively, empty containers shipped from  $i$  to  $j$ . Let  $c_i$  and  $d_i$  be the number of empty containers leased, respectively, stored at port  $i \in N$ . The profit function is then calculated as follows:

$$\begin{aligned} & \sum_{i \in N'} \sum_{j \in N'} a_{ij} P_{ij} - \sum_{i \in N'} \sum_{j \in N'} a_{ij} (L_i^f + U_j^f) - \sum_{i \in N'} \sum_{j \in N'} b_{ij} (L_i^e + U_j^e) \\ & - \sum_{i \in N'} (L_i c_i + S_i d_i) - \sum_{i \in N_{out}} F_i - \sum_{i \in N_{in}} F_i - K_0. \end{aligned} \quad (1)$$

The first term denotes the revenue collected for shipping the full containers, which is followed by the *cargo-handling cost* (that consists of: loading/unloading cost for full and empty containers, respectively, and cost for storing/leasing of empty containers), and *port call* costs. Finally,  $K_0$  denotes the fixed cost which is the sum of *capital cost* (the cost of the charter, including maintenance, insurance, crew) and the *bunker* (fuel) cost for the whole route. The value of  $K_0$  is calculated as follows (see [19]):

$$K_0 = (\text{dcc} \cdot T_{\max} + \text{P}_{\text{out}} \cdot (T_0 - T_w) \cdot (\text{fp} \cdot \text{scf} + \text{lp} \cdot \text{scl})) \quad (2)$$

Note that, for barge container shipping,  $K_0$  is a constant, as the length 1 is simply the distance between 1 and  $n$ , and it does not depend on the particular subset of calling parts. On the contrary, in the maritime

shipping, the route length can vary, and has to be calculated based on the sequence of calling ports (we address this issue in Section 5.4).

To calculate the total turnaround time, we have to take into consideration the time for loading and unloading full and empty containers at the respective ports, the time for arrival and departure and the calling ports, and the fixed time  $T_0$ :

$$\sum_{i \in N'} \sum_{j \in N'} a_{ij} (T_i^l + T_j^u) + \sum_{i \in N'} \sum_{j \in N'} b_{ij} (\tilde{T}_i^l + \tilde{T}_j^u) + \sum_{i \in N_{out}} (T_i^a + T_i^d) + \sum_{i \in N_{in}} (T_i^a + T_i^d) + T_0 \quad (3)$$

The value of  $T_{\max}$  is normally fixed to a multiple of 7 days, i.e., number of weeks. Note that, for the given line that is operated periodically (i.e., on a weekly basis), the value of  $T_{\max}$  implicitly determines the size of the fleet, i.e., for  $T_{\max} = 28$  and the weekly schedules, a fleet of four ships is needed to guarantee the service. Notice also that the total turnaround time is sometimes bounded from below by the value of  $T_{\min}$ . Usually, the value of  $T_{\min}$  is close to  $T_{\max}$  (e.g.,  $T_{\min} = T_{\max} - 1$ ) in order to avoid idle days for the ship, or waiting at the initial port that can be very expensive for the shipping company.

Finally, observe that the profit calculated by (1) is *weekly profit* for the shipping company for the *whole fleet*. Consider an optimal route whose optimal solution value is  $P$  and let the calculated turnaround time for this solution be  $w$  weeks. The weekly profit per ship is then  $P/w$ . However, to provide a regular service on the weekly basis, the company will have to employ a fleet of  $w$  ships, so that the total weekly profit for the company is  $P$ .

### 3.2. Transformation of the Input Graph

To simplify the notation and the description of our models, we introduce the directed acyclic graph (DAG)  $G = (\bar{N}, A)$  which is constructed by doubling all the ports (except the last one), so that the first (resp., second) copy corresponds to the (possible) visit of the port in the outbound (resp. inbound) direction. The set of arcs  $A$  ensures that only outbound, resp. inbound container shipping is allowed, but no containers can be carried in both directions. We have the following notation:

- The set of nodes  $\bar{N} = \{1, 2, \dots, n, n+1, n+2, \dots, 2n-1\}$  is an ordered set of  $2n-1$  nodes such that the nodes  $i \in \{1, \dots, n\}$  correspond to the outbound visit of port  $i$ , whereas nodes  $i \in \{n+1, \dots, 2n-1\}$  correspond to the inbound visits of ports  $\bar{i} = 2n - i$ . In the following we will use a mapping  $\bar{i} = 2n - i$  to refer to the physical port  $\bar{i}$  associated to the node  $i \in \bar{N}$ , whenever  $i > n$ . To each node  $i \in \bar{N}$ , we associate:

- $\bar{T}_i$  is the time necessary to visit port  $i$ . It is defined as:

$$\bar{T}_i := \begin{cases} T_i^a + T_i^d, & \text{if } i \leq n, \\ T_i^a + T_i^d, & \text{otherwise} \end{cases} \quad i \in \bar{N}$$

The definition of all other parameters ( $F_i, U_i^e, L_i^e, U_i^f, L_i^f, T_i^l, T_i^u, \tilde{T}_i^l, \tilde{T}_i^u$ ) is straightforwardly extended from set  $N$  to  $\bar{N}$ , namely, for  $i \leq n$ , the values remain unchanged, and for  $i > n$ , they are set to the respective value for port  $\bar{i} = 2n - i$ .

- Two nodes  $i$  and  $j$  from  $\bar{N}$  are connected by an arc  $a = (i, j) \in A$  iff  $i < j \leq n$  or  $n \leq i < j$ . To each arc  $(i, j) \in A$ , we associate the following parameters:

- $\bar{D}_{ij}$  is the weekly expected demand between  $i$  and  $j$ , and it is set as:

$$\bar{D}_{ij} := \begin{cases} D_{ij}, & \text{if } i < j \leq n, \\ D_{\bar{i}\bar{j}}, & \text{if } n \leq i < j \end{cases} \quad (i, j) \in A$$

- $\bar{P}_{ij}$  is the *net profit* for shipping a container from port  $i$  to port  $j$ , i.e., it is obtained by subtracting the container unloading and loading costs from the collected revenue:

$$\bar{P}_{ij} := \begin{cases} P_{ij} - U_j^f - L_i^f, & \text{if } i < j \leq n, \\ P_{\bar{i}\bar{j}} - U_j^f - L_i^f, & \text{if } n \leq i < j \end{cases} \quad (i, j) \in A$$

- $\bar{C}_{ij}$  is the cost per empty container shipped from  $i$  to  $j$ :

$$\bar{C}_{ij} := \begin{cases} L_i^e + U_j^e, & \text{if } i < j \leq n, \\ L_i^e + U_j^e, & \text{if } n \leq i < j \end{cases} \quad (i, j) \in A$$

- $\bar{T}_{ij}$ , resp.,  $\bar{T}_{ij}^e$ , is the sum of the loading and unloading time per full, resp., empty, container when shipped from  $i$  to  $j$ :

$$\bar{T}_{ij} := \begin{cases} T_i^l + T_j^u, & \text{if } i < j \leq n, \\ T_i^l + T_j^u, & \text{if } n \leq i < j \end{cases} \quad \bar{T}_{ij}^e := \begin{cases} \tilde{T}_i^l + \tilde{T}_j^u, & \text{if } i < j \leq n, \\ \tilde{T}_i^l + \tilde{T}_j^u, & \text{if } n \leq i < j \end{cases} \quad (i, j) \in A$$

Observe that  $\bar{P}_{ij}$  shall remain strictly positive, otherwise the O-D pair  $(i, j)$  can be removed from the set of demands (as the shipping company would normally not offer the service if these net profits are non-positive).

A feasible solution in the graph  $G$  can now be described using a subset of arcs  $A' \subset A$  such that each arc  $(i, j)$  is labeled using a tuple  $a_{ij}/b_{ij}$ , which means that  $a_{ij}$  full and  $b_{ij}$  empty containers are shipped from port  $i$  to port  $j$ . By construction, only shipping in the outbound, respectively, inbound direction is allowed. Nodes incident to  $A'$  define the calling ports, and the route can be automatically reconstructed by following the sequence of incident nodes in the outbound and then inbound direction. In the following we provide two examples to illustrate the basic concepts of the empty container repositioning.

*Example:*. Let us assume that we are given  $n = 4$  ports, such that the demands for transporting full containers (after the transformation of the input graph, as described above) are:  $\bar{D}_{12} = 2$ ,  $\bar{D}_{13} = 5$ ,  $\bar{D}_{34} = 7$ ,  $\bar{D}_{46} = 4$ ,  $\bar{D}_{57} = 3$ ,  $\bar{D}_{67} = 7$ . Let us furthermore assume that the ship capacity is  $C = 10$ , so that in a feasible solution all demands can be satisfied. Let  $\bar{T}_i = 1$  for all  $i \in \bar{N}$  and  $\bar{T}_{ij} = 0$ ,  $\bar{T}_{ij}^e = 0$ , for all  $(i, j) \in A$ . Let  $T_{\min} = 7$  and  $T_{\max} = 8$  (hence, a solution in which all four ports are called in both directions is feasible). In the following, we illustrate two feasible solutions, each of them corresponding to a route 1-2-3-4-3-2-1. However, in the first one (depicted in Figure 2), the balancing of containers is done by storing and leasing empty containers at respective ports, whereas in the second example (Figure 3), storage and leasing costs are avoided by transporting the empty containers along the route. Depending on the costs required for storage/leasing, one solution can be better than the other.

We use Figure 2 to explain the balancing of empty containers: at each port  $i \in N$ , *flow-balance constraints* have to be satisfied. So, for example, at port 2, there are 2 full containers unloaded in the outbound direction, there are 4 more unloaded in the inbound direction. There are zero containers loaded in the outbound, and 7 containers loaded in the inbound direction. Hence, the total difference between the unloaded and loaded containers in both directions is  $(7 + 0) - (2 + 4) = 1$ , and we conclude that one empty container has to be leased at port 2. Similarly, whenever this difference is negative, the corresponding number of containers has to be stored at the given location.

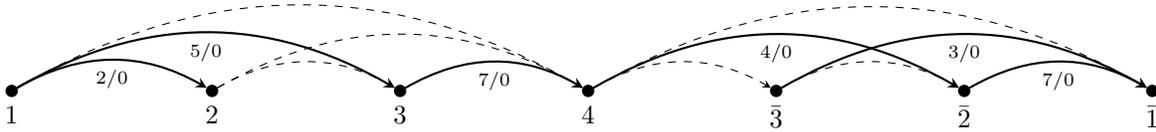


Figure 2: Example of a solution in which storage of empty containers is needed at port 1 (3 containers) and port 4 (3 containers), and leasing is necessary at port 2 (1 container) and port 3 (5 containers).

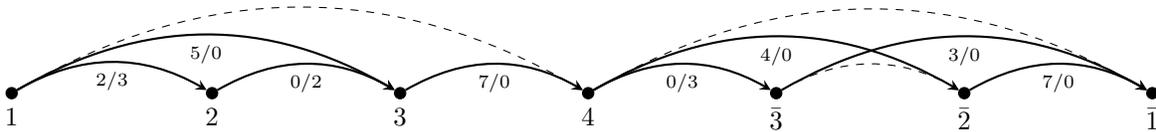


Figure 3: Example of a solution in which no leasing/storage of empty containers is needed, since the balancing is guaranteed by the shipping of empty containers.

### 3.3. NP-hardness

The BCSP contains a constraint associated to the upper bound on the total turnaround time. This is a knapsack-type constraint, so it follows that the BCSP is at least weakly NP-hard. We refer the interested

reader to [20], [13] for further details on the knapsack problem. In the following, we show two results: (1) we prove that the problem is in fact *strongly NP-hard*, even if most of the constraints are relaxed and the knapsack constraint is kept, and (2) in the case that the knapsack constraint and the ship-capacity constraint are relaxed, the problem can be solved in polynomial time. We say that the input instance is *capacity-unconstrained* if the ship capacity  $C$  is sufficiently large so that at every leg, complete demand can be shipped, i.e., if  $\sum_{(i,j) \in A: i \leq i', j > i'} \bar{D}_{ij} \leq C$  for each port  $i' \in \bar{N}$ . Similarly, we say that the instance is *time-unconstrained*, if the imposed interval  $[T_{\min}, T_{\max}]$  for the turnaround time is such that  $T_{\min} = T_0$  and  $T_{\max}$  is sufficiently large so that all ports can be called in both directions and all demands can be served.

The decision problem associated with BCSP consists of determining if there exists a solution ensuring a given profit.

**Theorem 1.** *The decision problem associated with BCSP is strongly NP-complete even if the input instance is:*

- *capacity-unconstrained,*
- *all costs are equal to zero (i.e.,  $\bar{C}_{ij} = 0$ , for all  $(i, j) \in A$ ),*
- *all demands and profits are binary (i.e.,  $\bar{D}_{ij}, \bar{P}_{ij} \in \{0, 1\}$ , for all  $(i, j) \in A$ ),*
- *$\bar{T}_i = 1$ , for all  $i \in \bar{N}$ ,  $T_0 = T_{\min}$ ,*
- *all leasing and storage costs are equal to zero (i.e.,  $L_i = S_i = 0$  for all  $i \in N$ ) and*
- *all loading and unloading times are zero.*

*Proof.* We will prove this result by reduction from the (decision variant of the) CLIQUE problem. Let  $H = (V, E)$  be an undirected graph,  $V$  the set of nodes,  $E$  the set of edges and let  $k$  be an integer. The decision variant of the CLIQUE problem consists of deciding if a subset of nodes  $Q \subseteq V$  of cardinality  $k$  exists such that the induced subgraph  $H[Q]$  is complete. We transform this instance of the CLIQUE problem into an instance of the BCSP in the following way. Without loss of generality we can order the nodes as follows:  $V = \{2, \dots, n-1\}$ . We build the DAG  $G = (\bar{N}, A)$  where  $\bar{N} = \{1\} \cup V \cup \{n\} \cup \bigcup_{i=2}^{n-1} \{\bar{i}\} \cup \{2n-1\}$  and  $A = \bigcup_{i=2}^n \{(1, i), (n, \bar{i})\} \cup \bigcup_{i=2}^{n-2} \bigcup_{j=i+1}^{n-1} \{(i, j), (\bar{i}, \bar{j})\} \cup \bigcup_{i=2}^{n-1} \{(i, n), (\bar{i}, 2n-1)\}$

Profits and demands are defined as follows:

$$\bar{P}_{ij} = \bar{D}_{ij} = \begin{cases} 1, & \text{if edge } ij \in E \vee \bar{i}\bar{j} \in E \vee i \in \{1, n, 2n-1\} \vee j \in \{1, n, 2n-1\} \\ 0, & \text{otherwise} \end{cases} \quad (i, j) \in A,$$

We set  $\bar{T}_i = 1$  for each port  $i \in \bar{N}$  and  $T_{\min} = T_0$  and  $T_{\max} = T_0 + 2k + 3$ . Figure 4 illustrates the transformation from the graph  $H$  into the DAG  $G$ , where dashed arcs correspond to the arcs where the

associated net profits and demands are equal to zero ( $\bar{P}_{24} = \bar{D}_{24} = \bar{P}_{25} = \bar{D}_{25} = \bar{P}_{52} = \bar{D}_{52} = \bar{P}_{42} = \bar{D}_{42} = 0$ ).

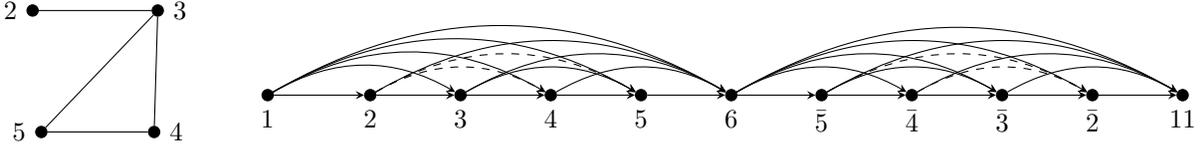


Figure 4: Transformation from CLIQUE problem into the BCSP on the DAG  $G$ .

Under the assumptions stated in the theorem, we observe that the optimal solution of the BCSP has a value  $(k + 2)(k + 1)$  if and only if the selected ports in this solution correspond to a clique of size  $k$  in  $G$ . Indeed, given the interval  $[T_{\min}, T_{\max}]$  for the turnaround time, at most  $2k + 3$  ports can be called (including the first and the last port), by any feasible BCSP solution. Observe that a profit between ports  $i$  and  $j$  (in the inbound and outbound direction) can be collected only if there exists an edge  $ij \in E$ . Hence, the total profit collected by traversing from 1 to  $n$  is at most  $\frac{(k+2)(k+1)}{2}$ , and the same holds for the profit collected from  $n$  to 1. If the induced subgraph defined by the visited ports is not complete, then the solution value is strictly less than  $(k + 2)(k + 1)$ . This also holds if less than  $k$  ports are visited between 1 and  $n$ . This completes the proof.  $\square$

**Corollary 1.** *The BCSP is strongly NP-hard.*

**Theorem 2.** *The BCSP is polynomially solvable in the restricted case in which:*

- *the instance is capacity-unconstrained,*
- *the instance is time-unconstrained, and*
- *leasing and storage costs for empty containers are equal to zero (i.e.,  $L_i = S_i = 0$ , for all  $i \in N$ ).*

*Proof.* We will prove this result by modeling the problem using two sets of binary variables: Let binary variables  $x_i$  be set to one iff port  $i$  is called,  $i \in \bar{N}$ , and, for each  $(i, j) \in A$ , let binary variables  $z_{ij}$  be set to one iff the complete demand  $\bar{D}_{ij}$  is satisfied. Due to the zero costs for leasing or storing empty containers, we easily observe that there always exists an optimal solution in which no empty containers need to be shipped. In this case the problem can be modeled as follows:

$$\max \sum_{(i,j) \in A} \bar{P}_{ij} \bar{D}_{ij} z_{ij} - \sum_{i \in \bar{N}} F_i x_i \quad (4)$$

$$z_{ij} \leq x_i \quad (i, j) \in A \quad (5)$$

$$z_{ij} \leq x_j \quad (i, j) \in A \quad (6)$$

$$x_i, z_{ij} \in \{0, 1\} \quad (i, j) \in A \quad (7)$$

Validity of this formulation follows from the fact that, if port  $i$  is called, all its demand will be covered (since there are no capacity restrictions and a solution in which the demand is partially fulfilled can always be improved by increasing the served demand). We observe that the constraint matrix defined by (5)-(6) is totally unimodular, hence, solving the LP-relaxation of this problem already provides an integer solution, which concludes the proof.  $\square$

#### 4. New MIP models for the BCSP

In this section we propose two new MIP formulations for the BCSP. Our models are much sparser when compared to those known in the literature, both in terms of the required decision variables and the underlying constraints. As we will demonstrate in the computational section, these models also provide significantly tighter lower bounds when compared to the previous formulation given in [19].

The basic property exploited by the new formulations is the fact that the sequence of ports that could be called is known in advance, and hence the routing aspect of the underlying optimization problem can be completely avoided. More precisely, once the calling ports are known, the underlying route is automatically given. Hence, there is no need to use arc variables for describing the route, it suffices to focus on the decisions whether a port  $i$  is called or not, and to distinguish between the calls in the outbound and inbound direction. The following variables are common in our both models:

- $x_i$  are binary variables which are set to one iff port  $i$  is called,  $i \in \bar{N}$ ,
- $z_{ij}$  is the number of full containers shipped from  $i$  to  $j$ ,  $(i, j) \in A$ ,
- $s_i$  is the number of empty containers stored at port  $i$ ,  $i \in N$ ,
- $l_i$  is the number of empty containers leased at port  $i$ ,  $i \in N$ .

In the DAG  $G$ , for a given  $S \subset \bar{N}$ , let  $\delta^+(S) = \{(i, j) \in A : i \in S, j \notin S\}$  denote the set of outgoing arcs from  $S$ , and similarly  $\delta^-(S) = \{(i, j) \in A : i \notin S, j \in S\}$  the set of incoming arcs. In the special case, for  $S = \{i\}$  we will write  $\delta^+(i)$  and  $\delta^-(i)$ , respectively. In the following, for each node  $i' \in \bar{N}$ , with  $A_{i'}$  we denote the arc-cut between the predecessors of  $i'$  (including  $i'$ ) and all its successors:

$$A_{i'} = \{(i, j) \in A : i \leq i', j > i'\} \tag{8}$$

By summing up the number of all containers shipped through  $A_{i'}$ , we obtain the load of the ship between ports  $i'$  and  $i' + 1$ . Obviously, for each  $1 \leq i' < 2n - 1$ , we must ensure that the total load does not exceed capacity  $C$ .

Finally, we also have to specify the repositioning of empty containers. Modeling of this repositioning comprises the major difference between the two MIP models considered in this paper. The first model

keeps track of the number of empty containers shipped between any two ports, whereas for the second model we only keep track of the number of empty containers that arrive, respectively, leave each port.

#### 4.1. First Model with Arc-Variables for Empty Containers

In our first model, we use arc variables  $y_{ij}$  associated to each arc  $(i, j) \in A$  with the following meaning:

- $y_{ij}$  is the number of empty containers shipped from  $i$  to  $j$ ,  $(i, j) \in A$ .

The following MIP model, that will be denoted by  $M_1^S$ , is a valid formulation for the BCSP (notation  $S$  stands for *splittable demand*):

$$\max \sum_{(i,j) \in A} (\bar{P}_{ij} z_{ij} - \bar{C}_{ij} y_{ij}) - \sum_{i \in \bar{N}} F_i x_i - \sum_{i \in \bar{N}} (S_i s_i + L_i l_i) \quad (9)$$

$$z_{ij} \leq \bar{D}_{ij} x_i \quad (i, j) \in A \quad (10)$$

$$z_{ij} \leq \bar{D}_{ij} x_j \quad (i, j) \in A \quad (11)$$

$$\sum_{(i,j) \in \delta^+(i)} (z_{ij} + y_{ij}) \leq C x_i \quad i \in \bar{N} \quad (12)$$

$$\sum_{(j,i) \in \delta^-(i)} (z_{ji} + y_{ji}) \leq C x_i \quad i \in \bar{N} \quad (13)$$

$$\sum_{(i,j) \in A_{i'}} (z_{ij} + y_{ij}) \leq C \quad i' \in \bar{N} \setminus \{\bar{1}\} \quad (14)$$

$$\sum_{(i,j) \in \delta^+(i)} (z_{ij} + y_{ij}) - \sum_{(j,i) \in \delta^-(i)} (z_{ji} + y_{ji}) + \sum_{(\bar{i},j) \in \delta^+(\bar{i})} (z_{\bar{i}j} + y_{\bar{i}j}) - \sum_{(j,\bar{i}) \in \delta^-(\bar{i})} (z_{j\bar{i}} + y_{j\bar{i}}) = l_i - s_i \quad i \in N \quad (15)$$

$$T_{\min} \leq T_0 + \sum_{(i,j) \in A} (\bar{T}_{ij} z_{ij} + \bar{T}_{ij}^e y_{ij}) + \sum_{i \in \bar{N}} \bar{T}_i x_i \leq T_{\max} \quad (16)$$

$$s_i, l_i \geq 0 \quad i \in N \quad (17)$$

$$x_i \in \{0, 1\} \quad i \in \bar{N} \quad (18)$$

$$z_{ij}, y_{ij} \in \mathbb{Z}_+ \quad (i, j) \in A \quad (19)$$

In this model, the objective function given in (9) maximizes the difference between the *net profit* ( $\bar{P}$ ) obtained for shipping the full containers, and the remaining cost that is composed of the cost for loading and unloading empty containers, cost for entering the ports (note that they will be paid twice if the same port is visited in the outbound and inbound direction), and cost for balancing containers at each port. Constraints (10) and (11) guarantee that full containers can be shipped from  $i$  to  $j$  only if both ports  $i$  and  $j$  are called. In addition, they impose the number of shipped containers not to exceed the demand  $\bar{D}_{ij}$ . Constraints (12) and (13) state that the complete ship load to be delivered at (or shipped from, respectively) port  $i$  cannot exceed ship capacity  $C$ , and in addition, nothing can be transported to/from a port, if the port is not called.

Inequalities (14) are the *capacity constraints* associated to the maximal capacity of the ship  $C$ : they ensure that the load of the ship (concerning both empty and full containers) between each node  $i'$  and  $i' + 1$  does not exceed  $C$ . Balancing of empty containers is given by constraints (15), where we again use the notation  $\bar{i} = 2n - i$ . For a given port  $i \in N$ , we calculate the difference of all containers loaded at  $i$  (either in the inbound or outbound direction) and containers unloaded at  $i$  (again, either inbound or outbound). If this difference is positive, the shipping company has to lease as many containers at port  $i$ , otherwise, it will need to store them. By minimization of  $S_i s_i + L_i l_i$  in the objective function, at optimality we necessarily have for each  $i$ :  $l_i \geq 0$  and  $s_i = 0$ , or  $s_i \geq 0$  and  $l_i = 0$ , but not  $l_i > 0$  and  $s_i > 0$ . Finally, we impose the length of the round trip to be in the interval  $[T_{\min}, T_{\max}]$  with constraint (16). The nature of decision variables is defined by (17)-(19). We do not explicitly impose integrality on  $l_i$  and  $s_i$ , since whenever variables  $x$ ,  $z$  and  $y$  are integer, variables  $l$  and  $s$  will automatically take integer values.

Notice that in the model  $M_1^S$ , due to the construction of the DAG  $G$ , all the cargo (including empty containers) is unloaded at port  $n$ , and then a new cargo is loaded at the same port. Hence, our model does not explicitly prevent a solution in which empty containers are carried from a port say  $i < n$  all the way to the port  $n$ , and then back to the port  $j$ , if this results in a less expensive solution. However, in the objective function, one would have to pay unloading and loading of these containers at the port  $n$ , whereas in reality, they would remain on the ship. To model this (rather extreme) situation, one would need to add a correction term to the objective function. Let  $y_n^-$  be an integer variable modeling the number of empty containers kept on the ship at port  $n$ . To the objective function one has to add the correction term  $(L_n^e + U_n^e)y_n^-$  where the correct value of  $y_n^-$  is guaranteed by the constraints:

$$\sum_{(j,n) \in \delta^-(n)} y_{jn} \geq y_n^- \quad \text{and} \quad \sum_{(n,j) \in \delta^+(n)} y_{nj} \geq y_n^-$$

For the sake of simplicity and without loss of generality, we do not involve variable  $y_n^-$  in the rest of the paper.

#### 4.2. Second (Aggregated) Model with Node Variables for Empty Containers

In contrast to the full containers, where each of them has a pre-specified origin and destination (and hence, each  $\bar{D}_{ij}$  has to be considered as a separate commodity), empty containers can be seen as a *single commodity* that can be picked up and/or delivered at any port. We therefore do not need to explicitly state the exact amount of empty containers transported from port  $i$  to port  $j$ , but rather the amount of empty containers that leave, respectively enter, each port. We modify model  $M_1^S$  to derive a second model, denoted by  $M_2^S$  by replacing the  $y_{ij}$  variables with these two new sets of variables:

- $y_i^{in}$  is the number of empty containers unloaded at port  $i$ ,  $i \in \bar{N}$ ,
- $y_i^{out}$  is the number of empty containers loaded at port  $i$ ,  $i \in \bar{N}$ .

Given these variables, we have to slightly modify the objective function so that the costs for loading/unloading empty containers at each port are handled separately. The model  $M_2^S$  reads as follows:

$$\max \sum_{(i,j) \in A} \bar{P}_{ij} z_{ij} - \sum_{i \in \bar{N}} (F_i x_i + L_i^e y_i^{out} + U_i^e y_i^{in}) - \sum_{i \in N} (S_i s_i + L_i l_i) \quad (20)$$

$$z_{ij} \leq \bar{D}_{ij} x_i \quad (i, j) \in A \quad (21)$$

$$z_{ij} \leq \bar{D}_{ij} x_j \quad (i, j) \in A \quad (22)$$

$$\sum_{(i,j) \in \delta^+(i)} z_{ij} + y_i^{out} \leq C x_i \quad i \in \bar{N} \quad (23)$$

$$\sum_{(j,i) \in \delta^-(i)} z_{ji} + y_i^{in} \leq C x_i \quad i \in \bar{N} \quad (24)$$

$$\sum_{(i,j) \in A_{i'}} z_{ij} + \sum_{i \leq i'} (y_i^{out} - y_i^{in}) \leq C \quad i' \in \bar{N} \setminus \{\bar{1}\} \quad (25)$$

$$\sum_{(i,j) \in \delta^+(i)} z_{ij} + y_i^{out} - \sum_{(j,i) \in \delta^-(i)} z_{ji} - y_i^{in} + \sum_{(\bar{i},j) \in \delta^+(\bar{i})} z_{\bar{i}j} + y_{\bar{i}}^{out} - \sum_{(j,\bar{i}) \in \delta^-(\bar{i})} z_{j\bar{i}} - y_{\bar{i}}^{in} = l_i - s_i \quad i \in N \quad (26)$$

$$T_{\min} \leq T_0 + \sum_{(i,j) \in A} \bar{T}_{ij} z_{ij} + \sum_{i \in \bar{N}} (\bar{T}_i x_i + \bar{T}_i^l y_i^{out} + \bar{T}_i^u y_i^{in}) \leq T_{\max} \quad (27)$$

$$s_i, l_i \geq 0 \quad i \in N \quad (28)$$

$$x_i \in \{0, 1\} \quad i \in \bar{N} \quad (29)$$

$$z_{ij} \in \mathbb{Z}_+ \quad (i, j) \in A \quad (30)$$

$$y_i^{in}, y_i^{out} \in \mathbb{Z}_+ \quad i \in \bar{N} \quad (31)$$

Constraints (21)-(27) are the adaptation of constraints (10)-(16), respectively. In order to balance the empty containers, four additional constraints are needed. Constraints (32) enforce that the amount of empty containers unloaded at a specific port cannot exceed the surplus of empty containers cumulated in the previous ports. The meaning of constraints (33) is similar, but it concerns the empty containers loaded at port  $i$ .

$$\sum_{j < i} y_j^{out} - \sum_{j < i} y_j^{in} \geq y_i^{in} \quad i \in \bar{N} \quad (32)$$

$$\sum_{j > i} y_j^{in} - \sum_{j > i} y_j^{out} \geq y_i^{out} \quad i \in \bar{N} \quad (33)$$

Finally, constraints (34), together with the structure of the DAG  $G$ , ensure that the whole cargo that was carried outbound (including empty containers) is unloaded at port  $n$ . Correspondingly, constraints (35), together with the structure of the DAG  $G$ , guarantee that the cargo is loaded at port  $n$  (or later), to be carried inbound.

$$\sum_{i=1}^{n-1} y_i^{out} - \sum_{i=1}^{n-1} y_i^{in} = y_n^{in} \quad (34)$$

$$\sum_{i=n+1}^{2n-1} y_i^{in} - \sum_{i=n+1}^{2n-1} y_i^{out} = y_n^{out} \quad (35)$$

In contrast to model  $M_1^S$ , the validity of model  $M_2^S$  is less obvious, and this result will be shown in the following subsection.

#### 4.3. Equivalence of the Two Models

With the following theorem we prove two results: First, we show that the model  $M_2^S$  is a valid formulation for the BCSP (by providing a bijection of solutions between the first and the second model). Second, we also prove that the two formulations,  $M_1^S$  and  $M_2^S$ , have the same value of the LP-relaxation (in which case, we call the two models *equivalent*).

**Theorem 3.** *Every (fractional) solution  $(\bar{x}, \bar{z}, \bar{s}, \bar{l}, y_{ij})$  of model  $M_1^S$  can be transformed into a (fractional) solution  $(\bar{x}, \bar{z}, \bar{s}, \bar{l}, y^{in}, y^{out})$  of model  $M_2^S$  with the same objective value, and vice-versa. The linear transformation is given as:*

$$y_i^{in} = \sum_{(j,i) \in \delta^-(i)} y_{ji} \quad (36)$$

$$y_i^{out} = \sum_{(i,j) \in \delta^+(i)} y_{ij} \quad (37)$$

*Proof.* The transformation from solutions of  $M_1^S$  to  $M_2^S$  simply fixes the values of variables  $y^{in}$  and  $y^{out}$  by equations (36) and (37). The transformation from solutions of  $M_2^S$  to  $M_1^S$ , that also ensures (36) and (37), is trickier and will be explained at the end of the proof.

Observe that if (36) and (37) hold, we have equality of objective values for the two models, which follows from the definition of  $\bar{C}_{ij}$ , since

$$\begin{aligned} \sum_{(i,j) \in A} \bar{C}_{ij} y_{ij} &= \sum_{i \in \bar{N}} \sum_{(i,j) \in \delta^+(i)} (L_i^e + U_j^e) y_{ij} = \\ &= \sum_{i \in \bar{N}} L_i^e \sum_{(i,j) \in \delta^+(i)} y_{ij} + \sum_{i \in \bar{N}} U_i^e \sum_{(j,i) \in \delta^-(i)} y_{ji} = \sum_{i \in \bar{N}} (L_i^e y_i^{out} + U_i^e y_i^{in}), \end{aligned}$$

whereas all the other terms remain equal in the objective functions of the two models.

Now, let us focus on constraints. Observe that if (36) and (37) are satisfied, then obviously constraints (12), (13), (15) and (16) for  $M_1^S$  become constraints (23), (24), (26) and (27) for  $M_2^S$  and vice-versa.

Moreover, the cut capacity constraints (14) become constraints (25) and vice versa because

$$\begin{aligned}
\sum_{i \leq i'} (y_i^{out} - y_i^{in}) &= \sum_{i \leq i'} \left( \sum_{j > i} y_{ij} - \sum_{j < i} y_{ji} \right) \\
&= \sum_{i \leq i'} \left( \sum_{j: i < j \leq i'} y_{ij} + \sum_{j > i'} y_{ij} - \sum_{j < i} y_{ji} \right) \\
&= \sum_{i \leq i'} \sum_{j > i'} y_{ij} = \sum_{(i,j) \in A_{i'}} y_{ij}
\end{aligned}$$

as in the second line above, the first and the third summation cancel out (as each arc  $(i, j)$  with  $i < j < i'$  appears with a positive and a negative sign), so that what finally remains is the summation of the arcs with origin  $i \leq i'$  and destination  $j > i'$ .

Now to finish the proof, we need to complete the missing parts studying one transformation after the other.

(i) Transformation from solutions of  $M_1^S$  to  $M_2^S$ . It remains to show that constraints (32) and (33) are satisfied. Indeed, by using (36) and (37), we get

$$\sum_{i < i'} (y_i^{out} - y_i^{in}) - y_{i'}^{in} = \sum_{i < i'} \sum_{j > i} y_{ij} - \sum_{j \leq i'} \sum_{i < j} y_{ij} = \sum_{i < i'} \sum_{j > i'} y_{ij} \geq 0$$

which follows from the fact that each arc  $(i, j)$  such that  $i < j \leq i'$  appears twice in the summation on the left-hand side, once with a positive and once with a negative sign, so that what finally remains is the sum of arcs that start before  $i'$  and end after  $i'$ . Hence, (32) is satisfied. Similarly, the validity of (33) can be shown, as they practically boil down to the same  $\sum_{i < i'} \sum_{j > i'} y_{ij} \geq 0$ .

(ii) Transformation from solutions of  $M_2^S$  to  $M_1^S$ . It finally remains to show that from given  $y_i^{in}$  and  $y_i^{out}$ , one can find values of variables  $y_{ij}$  such that (36) and (37) are satisfied, i.e., this system of equations has a solution. First, observe the following property:

$$\sum_{i \in \bar{N}} y_i^{out} - \sum_{i \in \bar{N}} y_i^{in} = 0 \tag{38}$$

This holds as by summing (34) and (35) we get

$$\begin{aligned}
0 &= \left( \sum_{i=1}^{n-1} y_i^{out} - \sum_{i=1}^{n-1} y_i^{in} - y_n^{in} \right) + \left( - \sum_{i=n+1}^{2n-1} y_i^{in} + \sum_{i=n+1}^{2n-1} y_i^{out} + y_n^{out} \right) \\
&= \sum_{i=1}^{2n-1} y_i^{out} - \sum_{i=1}^{2n-1} y_i^{in}
\end{aligned}$$

We show that the values of  $y_{ij}$  can be obtained by solving a circulation problem on an *extended digraph* in which node demands/supplies are defined using the values of  $y_i^{in}$  and  $y_i^{out}$ . This extended graph is constructed starting from the original digraph  $G$ , by adding for each node  $i \in \bar{N}$  two nodes  $i^-$  and  $i^+$ , and two arcs  $(i^-, i)$

with a lower bound and capacity both equal to  $y_i^{in}$  (to ensure a flow of  $y_i^{in}$  units on that arc) and  $(i, i^+)$  with a lower bound and capacity both equal to  $y_i^{out}$  for the same reason. Then for each  $(i, j) \in A$ , we add an arc  $(i^+, j^-)$  with capacity  $C - \bar{z}_{ij}$ . In Figure 4.3 we show an example of the extended digraph for an instance with 4 ports. Solving the system of equations (36)-(37) is equivalent to finding a feasible circulation in this modified graph with supply/demands  $d_i := y_i^{in} - y_i^{out}$  on nodes  $i \in \bar{N}$ . A sufficient condition for finding a feasible flow on such a graph is that  $\sum_{i \in \bar{N}} d_i = 0$  (see [14], section 7.7) which is exactly our property (38). So, we can indeed find the  $y_{ij}$  satisfying (36)-(37) from the  $y_i^{in}, y_i^{out}$  values. This completes the proof.  $\square$

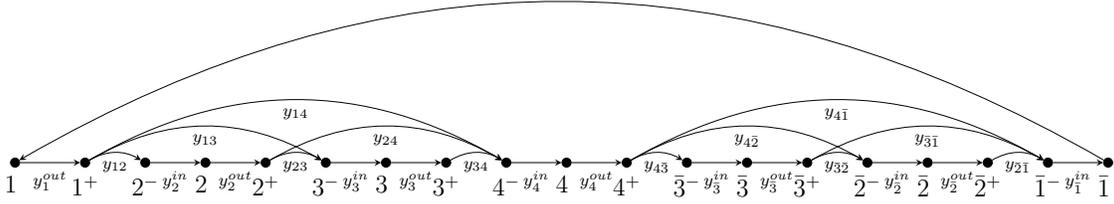


Figure 5: Example of extended graph for an instance of 4 ports

Even though the two formulations provide the same quality of lower bounds, it is not clear which one of them performs better from the computational point of view. This is because formulation  $M_2^S$  admits less decision variables, but more constraints when compared to  $M_1^S$ . On the one hand, formulation  $M_2^S$  strongly exploits the problem assumptions (outbound-inbound principle) and results into a “thinner” model. On the other hand, model  $M_1^S$  could be more flexible in terms of potential extensions concerning e.g., the time-dimension, simultaneous planning of multiple routes, or transshipment. Computational comparison of the two models, among other issues, will be investigated in Section 6.

#### 4.4. Properties of Optimal Solutions

We now introduce some properties of optimal solutions whenever special assumptions concerning cost, capacity or time limit parameters are satisfied.

**Proposition 1.** *If for each port  $i$ , we have  $U_i^e > L_i$ ,  $L_i^e > S_i$  and  $T_i^l + T_i^u < \bar{T}_i^l + \bar{T}_i^u$ , then all variables  $y_i$  will be zero at optimality and therefore can be removed from the model.*

*Proof.* Let us use model  $M_2^S$  for the proof. To balance containers at each port, constraints (26) can be rewritten as

$$\left( \sum_{(i,j) \in \delta^+(i)} z_{ij} + \sum_{(\bar{i},j) \in \delta^+(\bar{i})} z_{\bar{i}j} \right) - \left( \sum_{(j,i) \in \delta^-(i)} z_{ji} + \sum_{(j,\bar{i}) \in \delta^-(\bar{i})} z_{j\bar{i}} \right) = l_i - s_i + (y_i^{in} + y_i^{in}) - (y_i^{out} + y_i^{out})$$

which says that the difference between outflow and inflow at port  $i$  (computing flows at a port both outbound and inbound) should be exactly balanced by a mix of storing or leasing, and empty container repositioning. The right-hand side of the above flow balance equation has a corresponding cost of  $S_i s_i + L_i l_i + L_i^e (y_i^{out} + y_i^{out}) + U_i^e (y_i^{in} + y_i^{in})$  in the objective function (20). Therefore, if the cost assumptions of the proposition hold, balancing the containers with only storing or leasing (variables  $s_i$  or  $l_i$ ) without using any empty containers ( $y_i^{in} = y_i^{out} = 0$ ) will be less costly. Since the empty containers variables  $y_i^{in}$  and  $y_i^{out}$  consume ship capacity in constraints (23), (24), (25) and consume more time in constraint (27) if  $T_i^l + T_i^u < \bar{T}_i^l + \bar{T}_i^u$ , then these variables  $y_i^{in}$  and  $y_i^{out}$  will all be equal to zero at optimality.  $\square$

Consequently, to have an economic interest in transporting empty containers, we can assume that the conditions of Proposition 1 do not hold. We now introduce a second property of optimal solutions based on capacity and time-limit assumptions.

**Proposition 2.** *If (i) there is enough demand to fill the ship at any time (i.e.,  $\sum_{(i',j) \in A_i} \bar{D}_{i'j} \geq C$  for each port  $i \in \bar{N}$ ), and (ii) the instance is time-unconstrained, then the ship will carry full containers only and will be at full capacity  $C$  during the whole trip.*

*Proof.* Assume that  $\sum_{(i',j) \in A_i} \bar{D}_{i'j} \geq C$  for each port  $i \in \bar{N}$  and let  $(x^*, z^*, s^*, l^*, y^*)$  be an optimal solution such that there exists a port  $i \in \bar{N}$  such that  $\sum_{(i',j) \in A_i} z_{i'j}^* < C$ . In this solution, the ship might carry some empty containers (i.e., we might have  $\sum_{(i',j) \in A_i} y_{i'j}^* \neq 0$ ) without exceeding the overall capacity. Starting from this optimal solution, one can find a feasible solution  $(x^*, z', 0, 0, 0)$  that visits exactly the same ports, satisfies  $z'_{ij} \geq z_{ij}^*$  for all arcs  $(i, j)$  and  $\sum_{(i',j) \in A_i} z'_{i'j} = C$  for each  $i \in \bar{N}$ , by simply removing all empty containers and adding full containers up to systematically filling the ship capacity. In this modified solution, since the ship is always at full capacity with only full containers we have  $y_i^{in} = y_i^{out} = 0$  for each  $i \in \bar{N}$ . As the containers are already balanced by the  $z'$ -variables, we also have  $l_i = s_i$  for all  $i \in N$ . Moreover, as all profits satisfy  $\bar{P}_{ij} > 0$  and we added full containers to those already transported, the new solution  $(x^*, z', 0, 0, 0)$  has a strictly higher profit than the starting one, i.e.,  $\sum_{(i,j)} \bar{P}_{ij} z'_{ij} > \sum_{(i,j)} \bar{P}_{ij} z_{ij}^*$ , it has the same fixed costs associated to visited ports, and has zero leasing, storage, or empty container repositioning costs. So the objective value of  $(x^*, z', 0, 0, 0)$  is strictly better than that of  $(x^*, z^*, s^*, l^*, y^*)$ , which contradicts the fact that the optimal solution would not be at full capacity at each port.  $\square$

In practice, the total demand is often large enough to completely fill the ship at most of the segments. Therefore, the reason why the ship would not be at full capacity  $C$  is mainly the upper limit for the turnaround time ( $T_{\max}$ ), which implicitly bounds the amount of full containers to be shipped. Similarly, if the demand is not allowed to be split, there will more available capacity on each segment. This residual capacity at the ship is normally filled by empty containers, whenever this can bring savings with respect to leasing at storage costs. Both observations are verified in our numerical experiments, as we will see later (cf. Section 6).

## 5. Problem Variants

In the previous sections, our main focus was on modeling the profit maximization for a barge container shipping company, while taking into account estimated weekly demands and a given interval for the turnaround time. In this section we demonstrate that our models do not have to be limited to these special cases. We show the following:

- Instead of imposing the lower and upper bound limits for the turnaround time, our models can easily be adapted to find the *optimal turnaround time*, while at the same time maximizing the achieved (weekly) profit. Below we propose an adaptation of our models that applies for the case that the minimal turnaround time  $T_0$  is greater than a week.
- If the total route-length allows to schedule a container ship to perform multiple outbound-inbound routes within a week, then a natural question arises: how to increase the utilization of the ship so as to maximize the profit? The problem consists of finding an itinerary that can be scheduled multiple times within a week, so that the demands are partially collected within each roundtrip. This case has been addressed in [26] and below we demonstrate that with our models one can easily model this situation.
- Sometimes, it is not allowed to split demands, so that either 0 containers or all  $\bar{D}_{ij}$  containers have to be shipped, assuming ports  $i$  and  $j$  are called  $(i, j, \in \bar{N})$ . We show that our models can easily be modified to deal with this “unsplittable demand-case”.
- Finally, we show that our models are not limited to the barge-container shipping network design, but that they can be easily extended to the general maritime case in which the pre-ordering of ports is given, but the length of the whole route is not fixed.

To simplify the exposition, we explain necessary extensions starting from the model  $M_1^S$ , but similar modifications carry over to  $M_2^S$ .

### 5.1. Finding the Optimal Turnaround Time

Let us assume for a moment that the minimal turnaround time  $T_0$  requires at least one week to close the route. The lower and upper bound on the turnaround time ( $T_{\min}$  and  $T_{\max}$ , respectively) are often set by the decision maker based on some additional economical, operational or strategic constraints. It is not difficult to see that solutions in which a lower or an upper bound is imposed to the turnaround time may be suboptimal, and consequently, the shipping company may miss the opportunity to increase the profits and better utilize its fleet. Consider, for example, a solution that requires 3 weeks to close the route (with  $T_{\max}$  set to 3 weeks), so that the shipping company charters three ships in order to provide a regular service on a weekly bases. It may happen that by increasing  $T_{\max}$ , more time can be given to allow for a better

utilization of the ship, so that more containers can be shipped along the route. On the one hand, a route longer than 3 weeks implies that the company will charter additional ships to make sure the weekly service is provided. On the other hand, this may result in an increased profit per ship. Since the charter costs are already subtracted from the profit function (cf. constant  $K_0$ ), this implies that the overall annual gain of the shipping company increases. Similar arguments apply if the constraint that imposes the minimum turnaround time is relaxed. It is therefore worthwhile asking the question: can we simultaneously optimize the profit and determine the optimal turnaround time?

In what follows, we give a positive answer to this question and show how to adapt our previous model  $M_1^S$  so as to simultaneously determine the calling sequence, the shipping of full and empty containers and the optimal turnaround time. As customary, let us assume the demand is given on a weekly basis. We introduce two additional decision variables:

- $\lambda$ : an integer variable determining the optimal number of weeks for the turnaround time. Given that  $T_0$  is the minimal turnaround time (in days) needed to sail from 1 to  $n$  and back, we can calculate the lower bound on  $\lambda$  as  $\lambda_{\min} = \lceil \frac{T_0}{7} \rceil$ . Let  $\lambda_{\max}$  be a constant (given by the decision maker) determining the maximal acceptable number of weeks for the turnaround time.
- $\beta$ : the number of days the ship has to wait at the initial port, before starting the next route. Clearly, we can impose an upper bound on  $\beta$  to be  $\beta_{\max} \leq 6$  (but, normally, in practice,  $\beta_{\max} \leq 2$ ).

If a ship has to wait at the initial port, this may be costly for the shipping company. Let daily cost for staying at the port be  $C_\beta$ . The company wants to find the optimal turnaround time by taking these costs into account as well. Minimal changes to the model  $M_1^S$  are required. Daily charter costs now depend on the number of weeks the ship will be chartered, and we therefore replace  $\text{dcc} \cdot T_{\max}$  from the constant  $K_0$  by  $\text{dcc} \cdot 7\lambda$  in the objective function. The remaining (bunker) costs remain constant (we assume that the average ship speed will not be changed), and they are expressed by the constant  $K'_0$  below. In addition, we add the penalty term to the objective function ( $\beta C_\beta$ ), for the number of days the ship will have to wait at the initial port. Hence, the objective function (9) is replaced by:

$$\max \sum_{(i,j) \in A} (\bar{P}_{ij} z_{ij} - \bar{C}_{ij} y_{ij}) - \sum_{i \in \bar{N}} F_i x_i - \sum_{i \in N} (S_i s_i + L_i l_i) - \beta C_\beta - \text{dcc} \cdot 7\lambda - K'_0$$

where

$$K'_0 = P_{\text{out}} \cdot (T_0 - T_w) \cdot (\text{fp} \cdot \text{scf} + \text{lp} \cdot \text{scl}). \quad (39)$$

The profit function calculated above provides the *weekly profit* for the shipping company, assuming that a fleet of  $\lambda$  identical ships will be scheduled to ensure the weekly service.

Similarly, the time constraint (16) has to be replaced by the following ones (including the bounds on new variables):

$$\begin{aligned}
7\lambda - \beta &\leq T_0 + \sum_{(i,j) \in A} (\bar{T}_{ij} z_{ij} + \bar{T}_{ij}^e y_{ij}) + \sum_{i \in \bar{N}} \bar{T}_i x_i \leq 7\lambda \\
\lambda_{\min} &\leq \lambda \leq \lambda_{\max} \\
\lambda &\in \mathbb{Z}_+ \\
\beta &\in \{1, 2, \dots, \beta_{\max}\}
\end{aligned}$$

With these constraints we ensure that the turnaround time will be between  $\lambda$  weeks minus  $\beta$  days and  $\lambda$  weeks.

In our computational study (cf. Section 6) we study benefits of this model and demonstrate the gains in profit achieved when the turnaround time is optimized along with the route.

## 5.2. Multiple Round-trips

In the following, we study the complementary situation in which the container ship requires less than a week to close its route, so that it is theoretically possible to schedule multiple round trips using the same ship within a week, and the same route. In that case, our model requires similar adaptations as those proposed by [26]. We start with the model  $M_1^S$  and introduce an additional variable:

- $\alpha$ : the number of round trips made by the ship within a week. If at least one day is needed to sail from port 1 to port  $n$ , then  $\alpha \in \{1, 2, 3\}$ .

Since  $\alpha$  represents the number of round-trips the objective function must include the term  $\alpha \times K'_0$  where  $K'_0$  is defined as in (39), so the objective function reads as follows:

$$\max \sum_{(i,j) \in A} (\bar{P}_{ij} z_{ij} - \bar{C}_{ij} y_{ij}) - \sum_{i \in \bar{N}} F_i x_i - \sum_{i \in N} (S_i s_i + L_i l_i) - \alpha K'_0 - \text{dcc} \cdot T_{\max}$$

The inequalities (10), (11) and (16) have to be replaced by the following ones:

$$z_{ij} \leq \left\lfloor \frac{\bar{D}_{ij}}{\alpha} \right\rfloor x_i \quad (i, j) \in A \quad (40)$$

$$z_{ij} \leq \left\lfloor \frac{\bar{D}_{ij}}{\alpha} \right\rfloor x_j \quad (i, j) \in A \quad (41)$$

$$T_0 + \sum_{(i,j) \in A} (\bar{T}_{ij} z_{ij} + \bar{T}_{ij}^e y_{ij}) + \sum_{i \in \bar{N}} \bar{T}_i x_i \leq \frac{T_{\max}}{\alpha} \quad (42)$$

$$\alpha \in \{1, 2, 3\} \quad (43)$$

Constraints (40) and (41) ensure that if  $\alpha$  roundtrips are made, the full containers can be shipped multiple times between  $i$  and  $j$ , but the total number of them cannot exceed the demand  $\bar{D}_{ij}$ . Similarly, the time-constraint (42) guarantees that the length of  $\alpha$  round-trips does not exceed  $T_{\max}$  (which we assumed is a

week). The inequalities (40),(41) and (42) are quadratic. As it is done in [26], the simplest way to deal with these non-linearities is to run the model (at most) three times for different values of  $\alpha$ , and take the best result.

### 5.3. Modeling Unsplittable Demand

If it is not allowed to split the demand  $\bar{D}_{ij}$  between any two ports  $i$  and  $j$ ,  $(i, j) \in A$ , then our model  $M_1^S$  requires a slight modification, which consists of replacing  $z_{ij}$  by  $\bar{D}_{ij}w_{ij}$ , where the binary variable  $w_{ij}$  is set to one iff the complete demand  $\bar{D}_{ij}$  is shipped from  $i$  to  $j$ , i.e.:

$$w_{ij} = \begin{cases} 1 & \text{if demand from port } i \text{ to port } j \text{ is completely fulfilled} \\ 0 & \text{otherwise} \end{cases} \quad (i, j) \in A$$

In order to get a correct model for the unsplittable demands case, it is sufficient to replace every appearance of the variable  $z_{ij}$  in the model  $M_1^S$  by  $\bar{D}_{ij}w_{ij}$ , for all  $(i, j) \in A$ .

**Proposition 3.** *The value of the LP-relaxation of the model  $M_1^U$  ( $M_2^U$ ) is the same as the one obtained by the model  $M_1^S$  ( $M_2^S$ ).*

This result follows because every fractional feasible solution  $(\bar{x}, \bar{z}, \bar{s}, \bar{l}, \bar{y})$  for model  $M_1^S$  can be transformed into a fractional feasible solution  $(\bar{x}, \bar{w}, \bar{s}, \bar{l}, \bar{y})$  for model  $M_1^U$  with the same objective function value, and vice-versa using the linear transformation  $\bar{w}_{ij} = \frac{\bar{z}_{ij}}{\bar{D}_{ij}}$  for all  $(i, j) \in A$ .

Observe that all feasible integer solutions of  $M_1^U$  can be transformed to feasible integer solutions of  $M_1^S$  using this linear transformation, but trivially not vice-versa. Accordingly, the gap between the optimal integer solutions value of model  $M_1^U$  and its LP-relaxation value cannot be smaller than the LP-gap of model  $M_1^S$ . In our computational study (cf. Section 6), we computationally show that these gaps are considerably higher and accordingly, model  $M_1^U$  is harder to solve.

It is well known that unsplittable demands give less flexibility to the shipping companies, and hence lower ship utilization is normally achieved. This can be (partially) compensated by allowing the repositioning of empty containers as they can be used to fill the residual capacity of the ship, thereby saving the storage/leasing costs at the ports.

### 5.4. Designing Maritime Routes with Pre-specified Ordering of Ports

Let  $ST_{ij}$  be the Sailing Time from port  $i$  to port  $j$ . The underlying assumption for the constant sailing time  $T_0$  in the MIP formulations of Section 4 was that  $ST_{ij} = ST_{ik} + ST_{kj}$  for ports  $i < k < j$ ,  $(i, j) \in A$ , as it is the case for shipping along a river where the route necessarily passes by all ports. However, in the general case, see e.g. [26], [29], skipping an intermediate port  $k$  between  $i$  and  $j$  could strictly shorten the distance by an amount that is not negligible when it comes to the total bunker costs. We refer to this general

case where  $ST_{ij} < ST_{ik} + ST_{kj}$  for some triples  $((i, k, j), i < k < j, (i, j) \in A)$  as *shortcuts*. This can happen even in the restricted case of a pre-ordering of the sequence of ports that is studied all along this paper. In the following, we show how to adapt our models to deal with this general case even without introducing binary arc-variables to describe the route. The major difference to the BCSP is in the computation of  $K_0$  and  $T_0$  which are not constants anymore.

For each pair of ports  $(i, j) \in A$ , we introduce new continuous variables:

- $t_{ij}$  is equal to the travel time between ports  $i$  and  $j$  (for the chosen route), if ports  $i$  and  $j$  are two subsequently called ports on the route. Otherwise,  $t_{ij}$  is zero.

The following inequalities ensure the exact value of  $t_{ij}$ :

$$t_{ij} \geq ST_{ij}(x_i + x_j - 1 - x_k) \quad (i, j) \in A, i < k < j \quad (44)$$

$$t_{ij} \leq ST_{ij}x_i \quad (i, j) \in A \quad (45)$$

$$t_{ij} \leq ST_{ij}x_j \quad (i, j) \in A \quad (46)$$

$$t_{ij} \leq ST_{ij}(1 - x_k) \quad (i, j) \in A, i < k < j \quad (47)$$

$$t_{ij} \geq 0 \quad (i, j) \in A \quad (48)$$

Observe that the inequalities (45),(46) and (47) are redundant if the lower bound on the turnaround time ( $T_{\min}$ ) is not imposed.

These new continuous variables are now used to modify the objective function (9) and the time-constraint (16). We namely update the values of  $K_0$  and  $T_0$  as follows:

$$K_0 = dcc \cdot T_{\max} + P_{\text{out}} \cdot \left( \sum_{(i,j) \in A} t_{ij} \right) \cdot (\text{fp} \cdot \text{scf} + \text{lp} \cdot \text{scl})$$

$$T_0 = T_w + \sum_{(i,j) \in A} t_{ij}$$

Hence, the charter cost remain unchanged, whereas bunker cost are now calculated in function of the total distance traveled. Both, objective function and the turnaround constraints remain linear.

## 6. Computational Experiments

The goals of our computational study are as follows: (1) Evaluate the performance of the two formulations introduced in this paper and compare them with the state-of-the-art model from [19]; (2) Study the effects of rebalancing of empty containers on the achieved profits and the solution structure; (3) Measure how the empty container rebalancing is influenced by imposing splittable vs unsplittable demands (4) Test our

Table 2: The characteristics of 5 different container barge ships.

Container barge ships		No. units	TEU	$P_{out}$ [kW]	Total TEU	$v_1$ [km/h]	$v_2$ [km/h]	$d_{cc}$ [\$]
Ship 1	Motorized cargo push vessel	1	90	$2 \times 607$	215	19	11	3050
	Pushed barges	1	165	–				
Ship 2	Motorized cargo push vessel	1	145	$2 \times 1024$	409	20	12	4450
	Pushed barges	2	132	–				
Ship 3	Motorized cargo push vessel	1	77	$2 \times 565$	242	19	11	2850
	Pushed barges	1	165	–				
Ship 4	Motorized cargo push vessel	1	60	667	180	16	9	1750
	Pushed barges	2	60	–				
Ship 5	Motorized cargo push vessel	1	98	$2 \times 927$	338	20	12	3850
	Pushed barges	4	60	–				

models to determine optimal turnaround times, rather than imposing their lower and upper limits as hard constraints.

All the algorithms are coded in C/C++, and run single-thread on a PC with an Intel(R) Core(TM) i7-4770 CPU at 3.40GHz and 16 GB RAM memory, under Linux Ubuntu 14.04 64-bit. We used IBM-ILOG Cplex 12.6.0 (Cplex in the following) as a general-purpose MILP solver. All CPLEX parameters were set to their default values, except the following ones: relative and absolute tolerance were set to 0.0.

### 6.1. Benchmark Instances

We use the benchmark instances for the BCSP introduced in [19]: they consist of  $n$  ports, with  $n \in \{10, 15, 20, 25\}$ . In total, 20 instances are considered: for each value of  $n$ , five instances were produced with different ship characteristics (carrying capacities, daily charter costs, downstream and upstream speeds, engine outputs, fuel and lubricant consumptions, cf. Table 2). The real-world input parameters are taken from the *Container Liner Service Danube* project [7], where ports along the river Danube are taken as input. Other parameters are taken from [15, 17, 23, 25] and [24] and [28]. Benchmark instances are publicly available at [8].

### 6.2. Computational Performance

*Splittable Demands.* In the following, we consider the BCSP problem with splittable demands and compare the following three settings:

- $M_1^S$ : the first MIP formulation introduced in Section 4.1, based on arc-variables  $y_{ij}$  for modeling empty-container repositioning,

- $M_2^S$ : the second MIP formulation introduced in Section 4.2, based on node-variables  $y_i^{in}$  and  $y_i^{out}$  for modeling empty-container repositioning, and
- MLDM: the MIP formulation studied in [19].

All three models have been tested on the same machine whose features are described above. For the results of MLDM, we set a time limit of two hours.

Table 3 compares the three models in terms of the number of decision variables and constraints. Observe that it is sufficient to report a single line per each  $n \in \{10, 15, 20, 25\}$ , since the size of the models remains the same, once the number of ports is fixed. We notice that the MLDM model exhibits roughly twice as many variables as our new models and more than 50% more constraints. Comparing the size of the node-based model  $M_2^S$  with the arc-based one,  $M_1^S$ , we observe that the latter one contains about 50% more variables, whereas the number of constraints of the former one is slightly larger, but remains at the same scale as for  $M_1^S$ .

In Table 4 we compare the three models in terms of the following values: overall computing time in seconds (time), total number of branch-and-bound nodes (# nodes), LP-relaxation gap (lp gap) and final gap after reaching the time limit or proving optimality (exit gap). The LP-gap is defined as:

$$\text{lp gap} = \frac{LP - LB}{LB - K_0} \cdot 100\%,$$

where  $LP$  is the value of the LP-relaxation of the corresponding model, and  $LB$  is the best-known lower bound (or optimal solution). "Exit gap" is calculated as

$$\text{exit gap} = \frac{UB - LB}{LB - K_0} \cdot 100\%,$$

where  $UB$  is the global upper bound obtained upon the termination of the algorithm. Note that we report the gaps after removing the constant  $K_0$  from the objective function. Since in their original model, MLDM included  $K_0$  in the objective function, we also report its value in Table 4 for easier comparison with the results published in [19]. Column "lp gap" is reported only once for  $M_1^S$  and  $M_2^S$  (recall that the two models have the same quality of lower bounds). Finally, TL in the column "time" indicates that the time-limit was reached. Column OPT reports the optimal solution values.

The obtained results indicate that our new models are clearly superior to the MLDM formulation: with our models all benchmark instances are solved within seconds to optimality (in most of the cases within a fraction of a second), whereas for MLDM half of the instances could not be solved to optimality within two hours (with exit gaps ranging between 11% and 40%). This can be explained by two facts: (1) the size of the underlying formulations (cf. Table 3) and (2) by the quality of the LP-relaxations. Indeed, the LP-gap of the MLDM is as big as 180%, whereas our models exhibit an LP-gap which is consistently below 1% (with the exception of a single instance, for which the LP-gap is 1.9%). Consequently, relatively few branch-and-bound

Table 3: Number of variables and constraints of the different models for the splittable demands case.

instance	MLDM		Model $M_1^S$		Model $M_2^S$	
	# vars	# cons	# vars	# cons	# vars	# cons
Port10	358	398	219	246	167	286
Port15	758	818	479	521	327	581
Port20	1308	1388	839	896	537	976
Port25	2008	2108	1299	1371	797	1471

nodes are needed to prove optimality (hundreds, on average), whereas MLDM enumerates 3 to 4 orders of magnitude larger number of nodes to prove optimality (for  $n \in \{10, 15\}$ ) and for  $n \in \{20, 25\}$  it reaches the time limit after exploring hundreds of thousands of nodes.

Comparing the performance of  $M_1^S$  and  $M_2^S$ , no clear picture emerges: we may conclude that the models are competitive, both in terms of computing time and number of enumerated branch-and-bound nodes.

Finally, it is worth mentioning that even after running the MLDM model with a time limit of one day, most of the instances with  $n \in \{20, 25\}$  remained unsolved.

*Unsplittable Demands.* To have a closer look at the performance of the arc-based versus node-based model for the empty container repositioning, we also compare models  $M_1^U$  and  $M_2^U$ . Recall that the major difference with respect to  $M_1^S$  and  $M_2^S$  is that integer variables  $z_{ij}$  (denoting the number of full containers shipped from  $i$  to  $j$  in the splittable demand case) are replaced by  $\bar{D}_{ij}w_{ij}$  where binary variables  $w_{ij}$  are set to one iff complete demand  $\bar{D}_{ij}$  is shipped.

Table 5 compares the computational performance of  $M_1^U$  and  $M_2^U$ . The time limit for both models  $M_1^S$  and  $M_2^S$  was set to one hour. The following values are provided: global lower and upper bound upon the termination of the algorithm (LB and UB, respectively), and, as above, the total computing time in seconds, the number of branch-and-bound nodes, exit gap and the LP-relaxation gap (recall that also in this case, the quality of lower bounds of the two models is the same).

We first observe that imposing unsplittable demands appears computationally much more challenging, as only five out of 20 instances could be solved to optimality within the given time limit. This can be (partially) explained by the quality of LP-gaps: compared to the splittable-demand case, the LP-gaps for the unsplittable demands are much worse – they range between 1% and 6.4%. Hence, only for the smallest instances `Cplex` manages to close the gap within the time-limit, whereas for the remaining ones, even after exploring more than 1M of branch-and-bound nodes, the exit gaps remain around 0.1% - 0.5%. More importantly, the obtained results indicate that the sparser model, namely  $M_2^U$  consistently outperforms  $M_1^U$ :

Table 4: Computational performance of the three models for splittable demands.

instance	OPT	$K_0$	MLDM				Model $M_1^S$		Model $M_2^S$		
			time [s]	# nodes	lp gap [%]	exit gap	time [s]	# nodes	time [s]	nodes	lp gap [%]
Port10_1	110266.59	87927.58	2.43	3699	26.03	0.00	0.03	30	0.03	30	0.20
Port10_2	155604.92	130866.69	0.15	249	1.83	0.00	0.11	464	0.09	334	0.81
Port10_3	105370.17	82075.42	4.88	5821	31.88	0.00	0.04	84	0.04	63	0.14
Port10_4	73303.46	52617.19	0.42	591	8.76	0.00	0.13	297	0.09	108	1.90
Port10_5	140037.32	114722.33	0.84	1589	13.15	0.00	0.07	184	0.09	174	0.81
Port15_1	121546.54	109277.58	222.01	96914	146.79	0.00	0.03	0	0.03	0	0.02
Port15_2	187358.50	162016.69	85.31	54249	69.08	0.00	0.12	238	0.15	351	0.05
Port15_3	115823.64	102025.42	228.95	162075	158.94	0.00	0.06	0	0.06	13	0.03
Port15_4	87239.77	64867.19	897.73	523976	180.98	0.00	0.19	432	0.18	392	0.32
Port15_5	157472.29	141672.33	82.42	32528	101.04	0.00	0.08	12	0.05	10	0.01
Port20_1	129169.78	109277.58	TL	842898	136.64	22.95	0.30	90	0.26	177	0.26
Port20_2	195221.26	162016.69	TL	866050	66.81	11.60	0.31	137	0.20	99	0.08
Port20_3	123068.05	102025.42	TL	905052	148.36	23.97	0.86	530	0.71	562	0.45
Port20_4	92829.51	64867.19	TL	1064498	165.02	29.27	0.29	145	0.26	126	0.24
Port20_5	165930.22	141672.33	TL	805063	96.21	16.94	1.06	1919	1.10	2041	0.18
Port25_1	131101.13	109257.91	TL	403415	134.33	37.46	1.28	281	1.02	370	0.45
Port25_2	196427.43	162016.69	TL	363195	66.88	23.07	1.08	332	0.57	138	0.25
Port25_3	125469.35	102025.42	TL	401944	144.84	39.80	6.11	5777	5.57	5342	0.45
Port25_4	94044.71	64867.19	TL	453145	161.98	40.89	1.19	298	0.80	181	0.61
Port25_5	167862.13	141672.33	TL	416339	95.25	28.44	5.19	4272	4.50	4157	0.30

Table 5: Computational performance of models  $M_1^U$  and  $M_2^U$  for unsplittable demands.

instance	Model $M_1^U$					Model $M_2^U$					lp gap
	LB	UB	time	# nodes	exit gap	LB	UB	time	# nodes	exit gap	
Port10_1	<b>18204.52</b>	<b>18204.52</b>	2.34	2728	<b>0.00</b>	<b>18204.52</b>	<b>18204.52</b>	3.11	5067	<b>0.00</b>	4.10
Port10_2	<b>21872.31</b>	<b>21872.31</b>	0.42	1029	<b>0.00</b>	<b>21872.31</b>	<b>21872.31</b>	0.44	1557	<b>0.00</b>	2.71
Port10_3	<b>19783.36</b>	<b>19783.36</b>	2.15	2695	<b>0.00</b>	<b>19783.36</b>	<b>19783.36</b>	2.74	4099	<b>0.00</b>	3.59
Port10_4	<b>17608.09</b>	<b>17608.09</b>	0.18	189	<b>0.00</b>	<b>17608.09</b>	<b>17608.09</b>	0.15	213	<b>0.00</b>	6.37
Port10_5	<b>23297.05</b>	<b>23297.05</b>	0.85	1897	<b>0.00</b>	<b>23297.05</b>	<b>23297.05</b>	1.52	2667	<b>0.00</b>	2.28
Port15_1	<b>9474.87</b>	10291.53	TL	2247575	0.36	9471.80	<b>10166.69</b>	TL	6498917	<b>0.30</b>	2.37
Port15_2	<b>22706.26</b>	23054.53	TL	2603056	0.10	<b>22706.26</b>	<b>22849.24</b>	TL	5083504	<b>0.04</b>	1.48
Port15_3	<b>11574.65</b>	11791.13	TL	1243732	0.10	<b>11574.65</b>	<b>11741.77</b>	TL	4402257	<b>0.08</b>	1.99
Port15_4	19654.72	20544.58	TL	1103277	0.59	<b>19913.30</b>	<b>20287.72</b>	TL	4778290	<b>0.25</b>	3.23
Port15_5	<b>13561.16</b>	13945.35	TL	2059334	0.13	<b>13561.16</b>	<b>13767.15</b>	TL	6360404	<b>0.07</b>	1.45
Port20_1	<b>17270.88</b>	18001.00	TL	1076608	<b>0.31</b>	17160.03	<b>17959.15</b>	TL	2894820	0.34	2.34
Port20_2	29834.88	30975.79	TL	1344006	0.32	<b>30024.67</b>	<b>30959.53</b>	TL	2432769	<b>0.26</b>	1.74
Port20_3	18438.86	19674.42	TL	1477260	0.55	<b>18653.97</b>	<b>19575.81</b>	TL	3312351	<b>0.41</b>	2.44
Port20_4	24871.18	25972.45	TL	849980	0.71	<b>25022.21</b>	<b>25886.90</b>	TL	2891723	<b>0.56</b>	3.52
Port20_5	<b>21387.05</b>	22578.94	TL	1164365	<b>0.39</b>	21229.96	<b>22470.58</b>	TL	2774473	0.41	1.95
Port25_1	19317.20	20550.42	TL	990330	0.52	<b>19526.63</b>	<b>20349.93</b>	TL	1831172	<b>0.34</b>	2.25
Port25_2	32297.59	33059.46	TL	891725	0.21	<b>32334.48</b>	<b>33015.59</b>	TL	1909594	<b>0.19</b>	1.32
Port25_3	20726.61	22284.03	TL	1050004	0.69	<b>20878.69</b>	<b>22063.79</b>	TL	1977061	<b>0.52</b>	2.55
Port25_4	<b>25834.35</b>	<b>27248.33</b>	TL	785775	<b>0.90</b>	25763.13	27257.47	TL	1745961	0.95	4.32
Port25_5	23575.91	25009.42	TL	881915	0.46	<b>24015.08</b>	<b>24861.49</b>	TL	1495270	<b>0.27</b>	1.61

the best upper bounds are obtained with the  $M_2^U$  model, and consequently, the best exit gaps are reported with  $M_2^U$  in all but three cases. We therefore conclude that it indeed pays off to eliminate arc variables and model the empty container repositioning as a single commodity using node-variables only.

### 6.3. Analyzing Solutions: Splittable vs Unsplittable Demands

In the following, we analyze the structure of optimal solutions for both splittable and unsplittable demand cases. Splittable demand case allows a shipping company to adjust the number of containers accepted for loading and transportation in each port so as to achieve the highest value of profit. Unsplittable demand case is more oriented towards satisfaction of all customer requests in calling ports. Both profit and customer satisfaction are among the most significant business goals of any barge shipping company so these cases have its practical values and usefulness.

Tables 6 and 7 report the major solution features for the unsplitable and splitable-demand case, respectively. We report the number of ports in the calling sequence (outbound plus inbound) ( $\#calls$ ), the percentage of total demand fulfilled ( $\%D_1$ ) and the percentage of the total demand of the visited ports ( $\%D_2$ ). The average load (Avg. Load [%]) is calculated as the sum of the loads between every two consecutive ports, divided by the total number of ports. The number of ships required to fulfill the schedule is given in column “fleet” and corresponds to the turn-around time in weeks. Column “revenue” reports the total revenue  $\sum_{(i,j) \in A} P_{ij}z_{ij}$ , whereas column  $\bar{P}\bar{z}$  reports the net revenue after subtracting loading and unloading costs ( $\sum_{(i,j) \in A} \bar{P}_{ij}z_{ij}$ ). Fixed costs for the calling ports sequence are shown in  $\bar{F}\bar{x}$ . Costs for storing, leasing and loading and unloading empty containers are given in columns  $S\bar{s}$ ,  $L\bar{l}$  and  $\bar{C}\bar{y}$ , respectively.

By comparing the values of profits from Tables 6 and 7, we can see that the profit, in the case of splitable demands, is higher from 8% to 22 % compared to the unsplitable demand case. Therefore, it can be concluded that a barge shipping company should pay more attention to balancing the container flows and accepting the requests to the level that will enable higher profits, than to the needs to satisfy all requests from all customers in calling ports. Tendency to meet all the requirements not only decreases the profit but also becomes unrealistic in the view of constant growth on the market and a given capacity of the ship. This market situation was taken into account in our benchmark instances characterized with large transport demands.

In terms of the number of calling ports, percentage of all transport demand covered, percentage of transport demand covered at visited ports and utilization of ships carrying capacity, there are no significant differences in the results obtained for splitable and unsplitable demand cases. Particularly, the number of calling ports and calling sequences are the same or shows very small differences (at most one calling port is added or removed). Differences between these two cases in the percentage of covered total transport demand are mostly not greater than 2 % (just in one case it is more than 5 %). The results also show similar characteristics with regards to the percentage of transport demand covered at visited ports. These differences are also in the range (1-2 %) and just in one case the results differ 9 %. Transport demand covered at visited ports is around 60 % in average. Since transport demands in all instances are set at high level, the average utilization of carrying capacity is around 95 % for both splitable and unsplitable demand cases which can be considered as very satisfying from the shipping company point of view.

Having all these results in mind, we can see that there are no significant differences in calling sequences, transport demands covered and utilizations of carrying capacity for both cases. However, differences in achieved profits are significant (up to 22 %). The main reasons for this outcome are distinctions in freight rates and costs associated to each request for transport of containers. Therefore, these findings go in line with our claim that a barge shipping company should pay more attention to the characteristics of transport requests (such as number of containers to be transported, freight rate, transport and handling costs, empty

container repositioning costs, etc.), rather than striving to satisfy all customer requests.

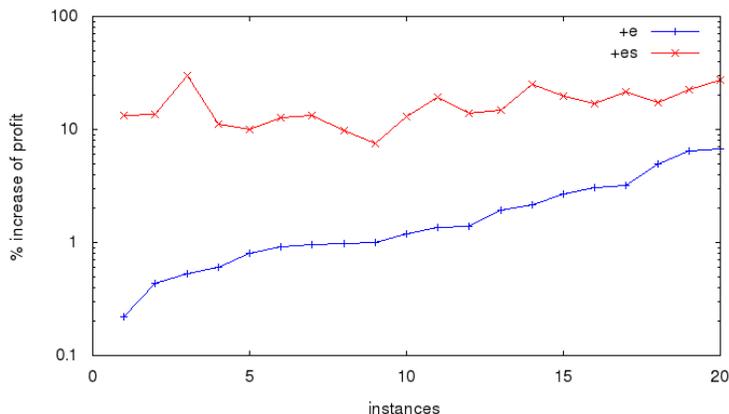
It is also obvious that empty container repositioning, storage and leasing costs are notably at low level, particularly if we take into account the total costs in one round trip. It was assumed that the shipping company does not charge a freight rate for transportation of empty containers. Therefore, the suggested models are trying to balance the number of containers transported to and from every port of call, which in turn leads to the reduction in demands for repositioning, storage and leasing of empty containers. Introducing the freight rates for transportation of empty containers, which is a more realistic case in barge transport, would increase both revenues and costs related to usage of empty containers. Therefore, in order to increase the flow of empty containers the proper values for these freight rates should be determined.

Obtained results for both demand cases (splittable and unsplittable) justify the importance of economies of scales in the shipping sector. It becomes obvious if we compare the TEU capacity of analyzed barge container ships, container demands and achieved profits for each instance. Ships with higher carrying capacity proved to be more profitable due to lower unit costs per TEU. These reduced costs come out since operating, voyage and capital costs do not increase proportionally with increase of TEU capacity of ships ([33, 19]). However, it is also necessary that customer demands and container flows among ports and terminals are large enough to ensure high utilization of carrying capacity of ships. This is the case with our benchmark instances. Results for instances with 10 ports, characterized with smaller total demands, are in line with this claim since highest profit is not achieved for the ship with largest TEU capacity. In instances with 15, 20 and 25 ports, barge container ship 2, with highest carrying capacity, reached the highest values of objective function.

As can be seen from the Tables 6 and 7, the results are obtained for the fleet size of 3 (instances with 10 ports) and 4 (instances with 15, 20 and 25 ports) ships. Since we consider weekly service, total turnaround time is limited to 21 (fleet size of 3) and 28 days (fleet size of 4). On the other hand, if we analyze the profits in both splittable and unsplittable demand cases, we can see that these values for instances with 10 ports, with smaller level of demands, are often higher compared to the profits for instances with 15, 20 and 25 ports. This can be explained by the importance of capital costs in the total shipping costs. It is known from the literature [16] that capital costs compose the largest portion of the total costs, and the obtained results clearly prove that. Furthermore, importance of capital costs is also confirmed by values of profit for the barge container ship 4. This ship has the smallest values of daily time charter costs, however, in instances with 15, 20 and 25 ports, for both demand cases, reaches the second best position.

Finally, we summarize the increase of profits that can be achieved by allowing splittable demands and/or empty container repositioning. In Chart 6 we start with the *basic setting* in which the demand cannot be split and no empty container repositioning is allowed (zero line). We then demonstrate: (1) the relative increase of profit (in %) if empty container repositioning is allowed (curve denoted by “+e”), and (2) the

Figure 6: Relative improvement of profit, compared to the setting in which loads cannot be split and no empty container repositioning is allowed (zero line).



relative increase of profit (in %) if both, empty container repositioning and splittable demands are possible (curve denoted by “+es”). Instances in this chart are sorted according to the increase with respect to “+e”. We observe that, only by shipping empty containers, the (weekly) profit (per ship) can be increased up to 7%, when compared to the basic setting. It is not surprising to see that the increase of profit is even more drastic when the demand can be split, in which case the profit obtained by the basic setting can be improved up to 30%.

#### 6.4. Optimizing Turnaround Time

Finally, we consider the situation in which the turnaround time is simultaneously optimized with the route through our model presented in Section 5.1. Table 8 shows the basic solution properties, but only for the instances for which we were able to obtain better solutions than those reported in Table 7. For 10 instances out of 25, we were able to improve the overall profits, by changing the turnaround time. More precisely, in all these cases, the optimal solutions are obtained by decreasing the turnaround time by one week. This clearly reduced the collected revenue, but increased the net profit, which can be explained by very high capital investments per ship.

## 7. Conclusion

In this article we studied the design of a route for a liner shipping company that provides regular service among the sequence of ports on a fixed-schedule basis. The models have been derived from the perspective of the shipping company that maximizes its revenue, given the estimated weekly demands, and under the assumption that the given ordering of ports has to be respected by the calling sequence. In contrast to the models considered in the previous literature, our models exploit the pre-ordering of ports in order to reduce

Table 6: Solution features for the unsplittable demand case.

instance	# calls	% d1	% d2	Avg. Load [%]	fleet	profit	revenue	$\bar{Pz}$	$F\bar{x}$	$S\bar{s}$	$L\bar{l}$	$\bar{C}\bar{y}$
Port10_1	13	40.92	75.40	90.6	3	18204.5	174032.0	113615.4	6772.0	0.0	0.0	711.3
Port10_2	15	49.20	72.88	80.2	3	21872.3	233562.0	161686.7	7906.0	391.0	457.6	193.1
Port10_3	13	40.77	75.74	95.2	3	19783.4	170125.0	109798.9	6732.0	176.0	194.9	837.3
Port10_4	8	21.64	82.08	87.3	3	17608.1	105774.0	75057.8	4110.0	206.6	130.0	385.9
Port10_5	14	50.51	81.62	89.1	3	23297.0	220107.0	145922.2	7286.0	115.7	148.3	352.9
Port15_1	27	40.78	46.24	95.9	4	9474.9	244266.0	133179.6	14104.0	0.0	0.0	323.2
Port15_2	28	53.01	56.29	94.5	4	22706.3	342562.0	200039.9	14656.0	67.5	70.6	522.9
Port15_3	27	41.48	47.03	95.2	4	11574.6	241215.0	128108.4	14104.0	0.0	0.0	404.3
Port15_4	23	30.45	45.74	97.8	4	19913.3	180726.0	97675.2	12043.0	155.7	228.2	467.8
Port15_5	28	48.97	52.00	96.0	4	13561.2	302618.0	170273.8	14656.0	35.3	35.3	313.7
Port20_1	28	28.37	52.37	96.3	4	17270.9	262488.0	141621.6	14681.0	164.1	175.0	53.1
Port20_2	28	33.91	59.65	94.4	4	30024.7	349461.0	207642.2	14872.0	0.0	0.0	728.8
Port20_3	28	28.37	50.74	96.4	4	18654.0	256486.0	136143.6	14792.0	0.0	0.0	672.2
Port20_4	24	20.93	49.07	96.1	4	25022.2	191879.0	102988.2	12381.0	99.7	141.1	477.0
Port20_5	29	32.17	53.98	95.6	4	21387.0	313981.0	178932.1	15253.0	76.4	66.6	476.6
Port25_1	27	21.22	56.54	97.3	4	19526.6	269328.0	143638.9	14370.0	0.0	0.0	484.4
Port25_2	28	24.20	63.22	96.6	4	32334.5	351103.0	209611.5	14910.0	43.7	56.3	250.4
Port25_3	28	20.82	54.56	96.6	4	20878.7	261113.0	138212.1	14881.0	30.8	34.7	361.5
Port25_4	23	15.31	53.77	96.3	4	25834.4	193535.0	103443.0	12201.0	0.0	0.0	540.4
Port25_5	28	23.64	62.24	96.7	4	24015.1	320228.0	181177.1	14970.0	92.7	160.4	266.5

Table 7: Solution features for the splittable demand case.

instance	# calls	% $D_1$	% $D_2$	Avg. Load [%]	fleet	profit	revenue	$\bar{P}\bar{z}$	$F\bar{x}$	$S\bar{s}$	$L\bar{l}$	$\bar{C}\bar{y}$
Port10.1	13	40.96	74.04	95.6	3	22339.0	177546.0	116958.6	6692.0	0.0	0.0	0.0
Port10.2	15	49.45	73.23	81.1	3	24738.2	236776.0	164146.0	7906.0	317.5	317.5	0.0
Port10.3	13	40.77	73.69	96.0	3	23294.7	172486.0	112062.2	6692.0	0.0	0.0	0.0
Port10.4	8	22.36	85.61	92.4	3	20686.3	109935.0	77678.5	4080.0	162.8	132.3	0.0
Port10.5	14	51.18	82.71	91.2	3	25315.0	222315.0	147442.4	7286.0	39.7	79.4	0.0
Port15.1	27	42.65	48.36	96.1	4	12269.0	251163.0	135650.5	14104.0	0.0	0.0	0.0
Port15.2	28	53.39	56.69	95.1	4	25341.8	345044.0	202014.5	14656.0	0.0	0.0	0.0
Port15.3	27	41.72	47.30	96.3	4	13798.2	243040.0	129927.6	14104.0	0.0	0.0	0.0
Port15.4	24	32.00	43.84	97.9	4	22372.6	187031.0	99682.8	12443.0	0.0	0.0	0.0
Port15.5	28	50.07	53.17	94.9	4	15800.0	306669.0	172128.3	14656.0	0.0	0.0	0.0
Port20.1	28	28.79	53.97	96.6	4	19892.2	265796.0	143810.8	14641.0	0.0	0.0	0.0
Port20.2	28	34.03	62.16	96.5	4	33204.6	352991.0	209924.3	14703.0	0.0	0.0	0.0
Port20.3	28	28.68	49.90	97.3	4	21042.6	258966.0	137459.1	14391.0	0.0	0.0	0.0
Port20.4	24	21.50	48.51	97.7	4	27962.3	196422.0	105320.5	12491.0	0.0	0.0	0.0
Port20.5	29	32.42	55.00	96.5	4	24257.9	317593.0	181183.2	15253.0	0.0	0.0	0.0
Port25.1	27	21.35	56.23	97.7	4	21843.2	271251.0	145392.1	14291.0	0.0	0.0	0.0
Port25.2	27	25.12	69.46	97.1	4	34410.7	358870.0	210827.4	14400.0	0.0	0.0	0.0
Port25.3	28	20.76	52.89	97.6	4	23443.9	263531.0	140280.4	14811.0	0.0	0.0	0.0
Port25.4	23	15.82	53.90	97.9	4	29177.5	199621.0	106155.7	12111.0	0.0	0.0	0.0
Port25.5	28	23.82	63.49	97.3	4	26189.8	323896.0	182842.1	14980.0	0.0	0.0	0.0

Table 8: Solution features for the optimal turnaround case. Only improved solution, when compared to those shown in Table 7 are reported.

instance	# calls	% $D_1$	% $D_2$	Avg. Load [%]	fleet	profit	revenue	$\bar{P}\bar{z}$	$F\bar{x}$	$S\bar{s}$	$L\bar{l}$	$\bar{C}\bar{y}$
Port15.1	14	23.84	75.64	94.3	3	24297.3	182722.0	119454.9	7230.0	0.0	0.0	0.0
Port15.2	15	28.58	80.97	85.5	3	30199.1	244266.0	168829.8	7764.0	0.0	0.0	0.0
Port15.3	14	23.71	75.21	94.7	3	25272.9	177790.0	114578.3	7230.0	0.0	0.0	0.0
Port15.5	14	29.70	85.78	91.4	3	29702.2	229485.0	151524.6	7100.0	0.0	0.0	0.0
Port20.1	13	15.22	85.66	94.5	3	24667.9	181735.0	119415.5	6820.0	0.0	0.0	0.0
Port20.3	14	14.57	77.33	94.9	3	25544.5	176111.0	114780.0	7160.0	0.0	0.0	0.0
Port20.5	14	18.85	85.59	91.5	3	30000.9	230490.0	151793.2	7070.0	0.0	0.0	0.0
Port25.1	13	10.89	85.93	95.1	3	24829.9	181742.0	119657.9	6920.0	0.0	0.0	0.0
Port25.3	14	10.29	77.11	96.4	3	25732.8	175028.0	115078.2	7270.0	0.0	0.0	0.0
Port25.5	14	13.54	85.59	90.5	3	30000.9	230490.0	151793.2	7070.0	0.0	0.0	0.0

the number of decision variables. In addition, we studied empty container repositioning between the ports and proposed two ways for balancing empty containers: the first model relies on arc-variables, the second model sees empty containers as a single-commodity and requires node-variables only.

In an extensive computational study on open benchmark barge shipping instances from the literature for which the optimal solution values were not known, we managed to prove the optimality within seconds. We also considered different scenarios and problem variants, and we proposed an effective way of incorporating them in our models. We finally analyzed the impact of these realistic variants on the achievable profits. The study has shown that: (i) by allowing empty container repositioning, better profits can be achieved in the unsplittable demand case (ii) with splittable demands the profits can be further (significantly) increased, and (iii) with letting our model simultaneously optimize the turnaround time and the route design, an additional increase of the profit can be achieved.

Concerning the future work, it remains to be studied how our models can be exploited for designing different routes for a fleet of ships (not necessarily homogeneous), and how to incorporate transshipment in the context of outbound-inbound shipping with empty container repositioning. While in this article we assume a constant speed all along the route, it is well known that speed optimization could allow for further increase of profits. This challenging setting involves a non-linear objective function, and an appropriate exact approach built on top of our model(s) remains to be investigated in the future.

## Acknowledgements

This work has been partially supported by Campus France, project number 36257TL and by Serbian Ministry of Education, Science, and Technological Development, grant numbers ON174033, TR36027 and a bilateral project no. 451-03-39/2016/09/09 from the Pavle Savić programme for years 2016/17. This support is greatly acknowledged.

## References

- [1] F. An, H. Hu, and C. Xie. Service network design in inland waterway liner transportation with empty container repositioning. *European Transport Research Review*, 7(2):1–11, 2015.
- [2] K. Braekers, A. Caris, and G. K. Janssens. Optimal shipping routes and vessel size for intermodal barge transport with empty container repositioning. *Computers in industry*, 64(2):155–164, 2013.
- [3] B. D. Brouer, D. Pisinger, and S. Spoorendonk. Liner shipping cargo allocation with repositioning of empty containers. *INFOR: Information Systems and Operational Research*, 49(2):109–124, 2011.

- [4] B. D. Brouer, J. F. Alvarez, C. E. Plum, D. Pisinger, and M. M. Sigurd. A base integer programming model and benchmark suite for liner-shipping network design. *Transportation Science*, 48(2):281–312, 2014.
- [5] M. Christiansen, K. Fagerholt, and D. Ronen. Ship routing and scheduling: Status and perspectives. *Transportation Science*, 38(1):1–18, 2004.
- [6] M. Christiansen, K. Fagerholt, B. Nygreen, and D. Ronen. Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, 228(3):467–483, 2013.
- [7] COLD. COLD, Container Liner Service Danube. Technical report, via Donau, ÖIR, Port of Constantza, Final Report, Vienna, 2006.
- [8] T. Davidovic. Routing of barge container ships by mixed-integer programming heuristics, 2016. URL [www.mi.sanu.ac.rs/tanjad/ships.htm](http://www.mi.sanu.ac.rs/tanjad/ships.htm).
- [9] J. Dong and D. Song. Container fleet sizing and empty repositioning in liner shipping systems. *Transportation Research Part E: Logistics and Transportation Review*, 45(6):860–877, 2009.
- [10] T. Glave, M. Joerss, and S. Saxon. The hidden opportunity in container shipping. *Mc Kinsey publications*, 2014. URL <http://www.mckinsey.com/business-functions/strategy-and-corporate-finance/our-insights/the-hidden-opportunity-in-container-shipping>.
- [11] Y.-F. Huang, J.-K. Hu, and B. Yang. Liner services network design and fleet deployment with empty container repositioning. *Computers & Industrial Engineering*, 89:116–124, 2015.
- [12] C. V. Karsten. *Competitive Liner Shipping Network Design*. PhD thesis, Technical University of Denmark, 2015.
- [13] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, Berlin Heidelberg, 2004.
- [14] J. Kleinberg and E. Tardos. *Algorithm design*. Pearson Education India, 2006.
- [15] R. Konings. Hub-and-spoke networks in container-on-barge transport. In *Transportation Research Record: Journal of the Transportation Research Board*, 1963, NRC, Washington, D.C.:23–32, 2006.
- [16] R. Konings. *Intermodal Barge Transport: Network Design, Nodes and Competitiveness*. PhD thesis, Technische Universiteit Delft, 2009.
- [17] R. Konings and H. Priemus. Terminals and the Competitiveness of Container Barge Transport. *Transportation Research Record: Journal of the Transportation Research Board*, 2062(1):39–49, 2008.

- [18] V. Maraš. Determining optimal transport routes of inland waterway container ships. *In Transportation Research Record: Journal of the Transportation Research Board*, 2062, NRC, Washington, D.C.:50–58, 2008.
- [19] V. Maraš, J. Lazić, T. Davidović, and N. Mladenović. Routing of barge container ships by mixed-integer programming heuristics. *Applied Soft Computing*, 13(8):3515–3528, 2013.
- [20] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Chichester, New York, 1990.
- [21] Q. Meng and S. Wang. Liner shipping service network design with empty container repositioning. *Transportation Research Part E: Logistics and Transportation Review*, 47(5):695–708, 2011.
- [22] Q. Meng, S. Wang, H. Andersson, and K. Thun. Containership routing and scheduling in liner shipping: Overview and future research directions. *Transportation Science*, 48(2):265–280, 2014.
- [23] T. Notteboom. Container river services and gateway ports: Similarities between the Yangtze River and the Rhine River. *Asia Pacific Viewpoint*, 48(3):330–343, 2007.
- [24] Port of Antwerp. Port of Antwerp, 2016. URL [www.portofantwerp.com](http://www.portofantwerp.com).
- [25] Z. Radmilović, B. Dragović, and V. Maraš. Power unit – cargo space link in transport. *YUJOR*, 15(1):50–58, 2005.
- [26] K. Rana and R. Vickson. A model and solution algorithm for optimal routing of a time-chartered containership. *Transportation Science*, 22(2):83–95, 1988.
- [27] K. Rana and R. Vickson. Routing container ships using lagrangean relaxation and decomposition. *Transportation Science*, 25(3):201–214, 1991.
- [28] Rotterdam Port Information. Rotterdam port information, 2016. URL [www.rotterdamportinfo.com](http://www.rotterdamportinfo.com).
- [29] K. Shintani, A. Imai, E. Nishimura, and S. Papadimitriou. The container shipping network design problem with empty container repositioning. *Transportation Research Part E: Logistics and Transportation Review*, 43(1):39 – 59, 2007.
- [30] D.-P. Song and J.-X. Dong. Cargo routing and empty container repositioning in multiple shipping service routes. *Transportation Research Part B: Methodological*, 46(10):1556–1575, 2012.
- [31] D.-P. Song and J.-X. Dong. Long-haul liner service route design with ship deployment and empty container repositioning. *Transportation Research Part B: Methodological*, 55:188–211, 2013.

- [32] D.-P. Song and J.-X. Dong. Empty container repositioning. In *Handbook of Ocean Container Transport Logistics*, pages 163–208. Springer, 2015.
- [33] M. Stopford. *Maritime Economics*. Routledge, Oxford, UK, 2009.
- [34] UNCTAD. United nations conference on trade and development: Review of maritime transport 2015, 2015. URL <http://unctad.org/en/pages/PublicationWebflyer.aspx?publicationid=1374>.
- [35] Z. Yang, H. Shi, K. Chen, and H. Bao. Optimization of container liner network on the yangtze river. *Maritime Policy & Management*, 41(1):79–96, 2014.

## Appendix: Schematic Representation of Optimal Solutions

In the following, we provide schematic representations for optimal solutions of the five instances with 10 ports for the BCSP with unsplittable demand. Drawings in Figures 7 to 11 show the optimal routes for five different barge container ships, respectively (cf. Table 2 for their basic characteristics). The solutions are defined by the upstream and downstream calling sequence and the number of loaded and empty containers transported between any two ports. Shaded nodes represent called ports, and the notation " $a + b$ " refers to the number of full and empty containers, respectively. The port  $P_1$  is at the river mouth, whereas port  $P_{10}$  is furthest port in the upstream direction.

Figure 7: Optimal route of barge container ship for 10 possibly calling ports and schematic overview of obtained container flows. Instance Port10\_1.

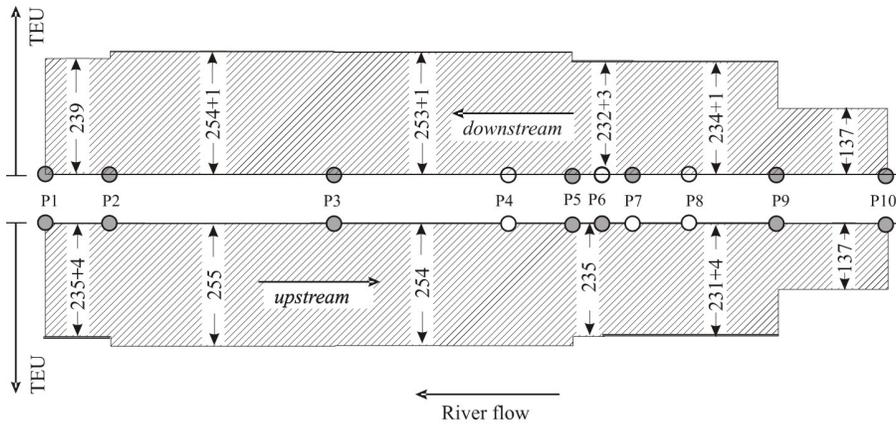


Figure 8: Optimal route of barge container ship for 10 possibly calling ports and schematic overview of obtained container flows. Instance Port10.2.

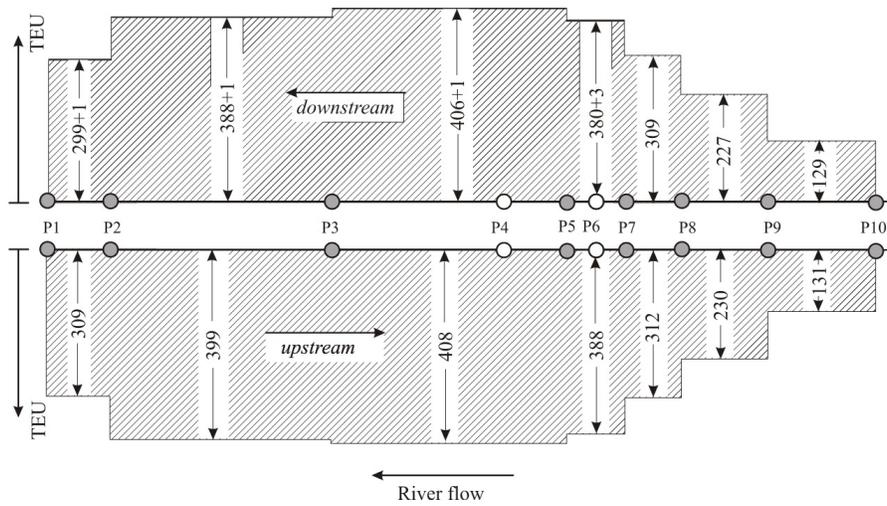


Figure 9: Optimal route of barge container ship for 10 possibly calling ports and schematic overview of obtained container flows. Instance Port10.3.

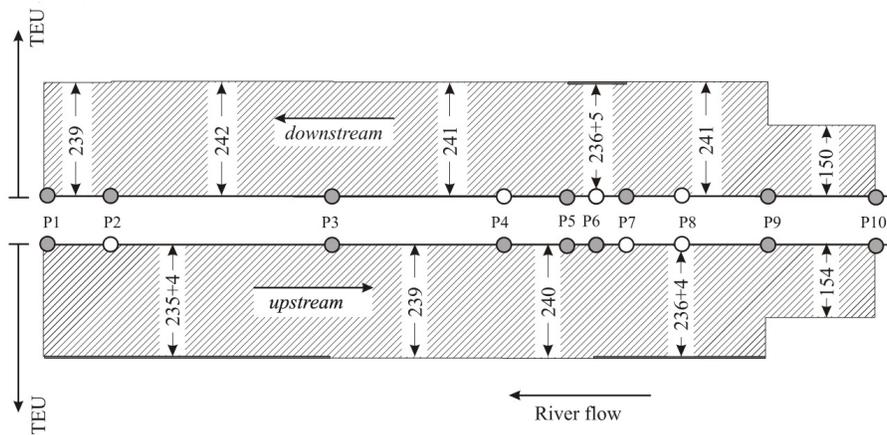


Figure 10: Optimal route of barge container ship for 10 possibly calling ports and schematic overview of obtained container flows. Instance Port10.4.

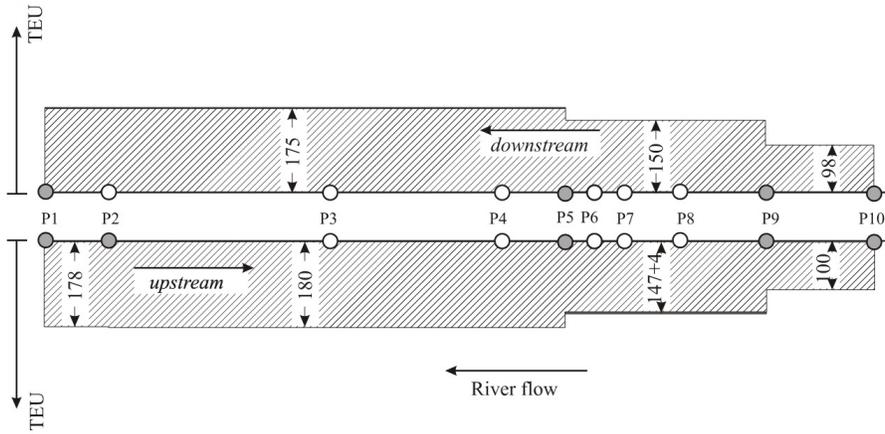


Figure 11: Optimal route of barge container ship for 10 possibly calling ports and schematic overview of obtained container flows. Instance Port10.5.

