Tighter MIP Models for Barge Container Ship Routing

Laurent Alfandari[a], Tatjana Davidović[b], Fabio Furini[c], Ivana Ljubić[a,*], Vladislav Maraš[d], Sébastien Martin[e]

[a] *ESSEC Business School, Cergy-Pontoise, France*
[b] *Mathematical Institute of the Serbian Academy of Sciences and Arts, Belgrade, Serbia*
[c] *PSL, Université Paris Dauphine, Paris, France*
[d] *University of Belgrade, Faculty of Transport and Traffic Engineering, Belgrade, Serbia*
[e] *LCOMS, Université de Lorraine, Metz, France*

Abstract

This paper addresses the problem of optimal planning of a liner service for a barge container shipping company. Given estimated weekly demands between pairs of ports, 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. The line has to follow the outbound-inbound principle, starting from the port at the river mouth. We propose a novel integrated approach in which the shipping company can simultaneously optimize the route (along with repositioning of empty containers), the choice of the final port, length of the turnaround time and the size of its fleet. To solve this problem, a new mixed integer programming model is proposed. On the publicly available set of benchmark instances for barge container routing, we demonstrate that this model provides very tight dual bounds and significantly outperforms the existing approaches from the literature for splittable demands.

We also show how to further improve this model by projecting out arc variables for modeling the shipping of empty containers. Our numerical study indicates that the latter model improves the computing times for the challenging case of unsplittable demands. We also study the impact of the turnaround time optimization on the total profit of the company.

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

[*] Corresponding author

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

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 multiple commodities. Due to the global financial crisis and the turmoil in global sea freight, the container shipping business is hardly profitable (see [12]). 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. 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 [5, 6, 8, 14, 24]). 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 which are typically considered in the barge container routing:

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 usually a sea port, located at a river mouth, see Figure 1).
3. There is no transshipment.
4. At the last visited port (in barge transport, it is the furthest visited port upstream), the ship changes sailing direction. This port is not known in advance.
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 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 various costs that include fuel cost, port dues and cargo handling cost.

According to [37], the containerized cargo flows on major container trade routes are characterized by huge imbalances between inbound and outbound directions, see also Song and Dong [35]. 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.

Liner shipping network design under Assumptions 1-3 (with an additional assumption that the final port is fixed) has been introduced in the seminal paper [29]. 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 simultaneously taking into account:

(a) empty container balancing and repositioning,

(b) optimal turnaround time (and, consequently, the size of the homogeneous fleet needed to operate the line), and

(c) optimal choice of the final port in the outbound direction.

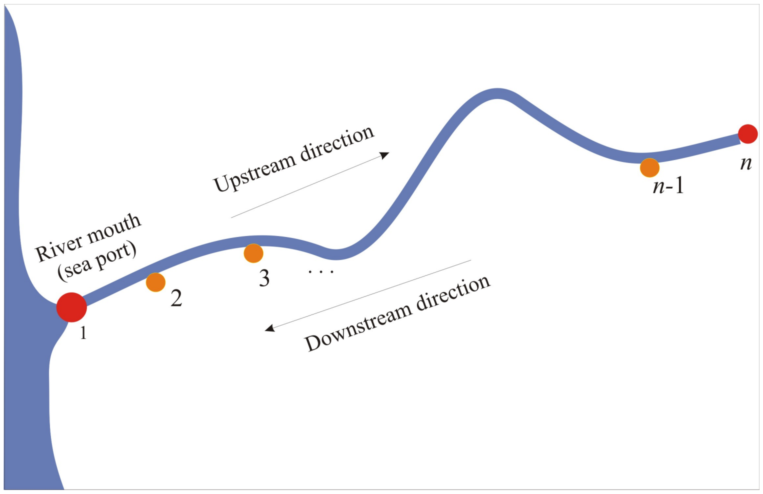


Figure 1: An example of liner shipping along a river with n ports.

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 [29]. The major assumption in this setting is that the ordering of ports for a given ship is predetermined. To our knowledge, an approach to design the optimal ship route involving empty container repositioning was considered for the first time in [32]. 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 [21].

Contributions. In this article, we propose an exact solution approach for barge container shipping that explicitly takes advantage of the outbound-inbound principle. To the best of our knowledge, this is the first exact approach for barge container shipping that simultaneously searches for the optimal route, while taking care of empty container balancing and repositioning, optimizing the turnaround time and the size of the (homogeneous) fleet, and searching for the final port along the route. In contrast to the usual utilization of arc-variables for modeling the routes, our approach exploits node-variables for the route design. Two formulations are given: the first formulation requires arc-variables for modeling empty containers, whereas in the second formulation these variables are projected out and replaced by a smaller set of node-variables that handle empty containers as a single commodity. An equivalence of the two models, concerning the strength of their linear programming relaxation bounds, is shown. 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.

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 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. Computational experiments on benchmark instances are given and analyzed in Section 5, whereas Section 6 concludes the paper.

2. Related Work

A classification of optimization problems for liner container ship routing was given in the recent surveys [6, 8, 24]. Following their classification, our problem falls into the category *Liner Container Shipping Network Design* (single route or several routes without transshipment). These three surveys, along with a recent paper [5], cite a dozen of papers published in the last decade in that specific category. An older survey [7] provides a list of papers on general ship routing and scheduling.

The recent article [5] 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 in [5] 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.

Recently three articles appeared in the maritime planning literature which deal with stochastic aspects, uncertainty factors or simulation. In [38], market uncertainties are considered and a two-stage stochastic model is proposed. In [11], a decision support methodology for strategic planning is presented using a Monte Carlo simulation framework. Finally in [25], the authors seek to design robust schedules of ships under weather uncertainty, and a simulation-optimization based methodology is presented. All these articles consider real-case instances.

General maritime route design with outbound-inbound principle and without transshipment. The previously cited paper of [29] falls into this category in which no transshipment is allowed, i.e., exchanging containers between two ships is not an option. In [30] 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 [29] 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 α. Although this leads to a quadratic model, the variable α can only take a few integer values, so that the authors propose to solve the problem by enumerating all possible values of α. This boils down to solving the same model for each α 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 [30] Lagrangian relaxation and decomposition is involved. In [19], the authors propose a branch-and-cut algorithm for finding shipping routes that respect pickup and delivery requests and take into consideration draft limits.

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 [32]. 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 [29]. 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 not applicable in inland waterway transport, 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 [32]. 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 [10], a multi-route multi-vessel problem is considered. In [23], the authors design a hub-and-spoke network with multiple routes. In [4], 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 [33], the authors consider a problem with muliple liner ships where demands between ports are to be satisfied while minimizing costs, including transhipment 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 [34], 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 [13]. The remaining papers from Table 1 deal with barge i.e., inland waterway liner transportation and will be addressed in the following paragraphs.

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 [29], 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 [29] is that the location of ports along a river induces a fixed travel time between the starting and last port. In addition, as in [32], the balancing and repositioning

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
[32]	×	×		$\times^a$	single
[21]	×	×		×	single
[3]	×	×		$\times^b$	multi
[2]	×	×		×	multi
[10]	×		×	$\times^c$	multi
[23]	×		$\times^d$		multi
[4]	×		×		multi
[33]	×		×		multi
[34]	×	×		$\times^e$	single^e
[13]	×		×		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. Inboud-outbound principle holds for each cycle.

of empty containers is considered. In [20], the author proposes a MILP formulation with binary variables associated with each pair of ports. This formulation, along with the MILP-based heuristics is implemented in [21] 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. Also, the last port of the route is fixed whereas leaving that as a decision to make, as we do in this paper, would enable more flexibility to increase profit further. Other papers specifically dealing with barge route design and inland waterway liner transportation with empty container repositioning are [2, 3, 39].

The three latter articles deal with multi-route multi-vessel optimization. The first one considers the selection of unsplittable demands that maximizes profit, whether the other two 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. Also note that none of these papers optimizes the turnaround time of the route.

3. Notation and Problem Definition

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

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

- $N = \{1, \ldots, n\}$: ordered set of n ports, where 1 is the starting port and n is the last port that can be visited in the outbound direction;
- $D_{ij} \in \mathbb{Z}_+$: weekly expected number of full containers available to be transported between ports i and j [TEU/week];
- $C \in \mathbb{Z}_+$: carrying 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]
- $w_{\min}$: minimum allowed turnaround time [weeks]
- $w_{\max}$: maximum allowed turnaround time [weeks]
- C_I : cost per time for keeping the ship idle at the initial port [US$/h]
- H_I : maximum allowed idle time at the initial port [h]
- T_i: total sailing time to go from port 1 to port i, and to go back to port 1, for each $i \in N$ [h]
- T_i^w : total waiting time for crossing borders and locks along the route from 1 to i in both directions, for each $i \in N$ (T_i^w is counted in T_i) [h].

Note that the sailing time T_i, for each $i \in N$, does not comprise the stopping times at ports for loading and unloading containers. We have $T_i \geq T_{i-1}$, for all $i \in N$, $i > 1$. The total travel time is the sum of the sailing time and the stopping time at ports.

Additional parameters are:

- `wcc` : weekly time charter cost of a ship [US$/week];

- $\mathtt{P}_{\text{out}}$: engine output (propulsion) [kW];
- `fp (lp)` : fuel (lubricant) price [US$/t];
- `scf (scl)` : specific fuel (lubricant) consumption [t/kWh];

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. Problem Definition

In this section we provide a formal definition of the Barge Container Shipping Problem.

Definition 1 (Barge Container Shipping Problem (BCSP)). *Given the input parameters described above, the BCSP asks to determine*

(i) the turnaround time w in weeks (which is also the size of the fleet),

(ii) the last port $i^* \in N \setminus \{1\}$ of the route,

(iii) the sequence of ports $N_{out} \subseteq N \setminus \{i^*+1, \ldots, n\}$ called in the outbound direction,

(iv) the sequence of ports $N_{in} \subseteq N_{out} \setminus \{i^*\}$ called in the inbound direction,

(v) the numbers z_{ij} and y_{ij} of full and empty containers, respectively, to be shipped between ports i and j,

(vi) the numbers l_i and s_i of empty containers leased, respectively stored at port $i \in N$,

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* (2)*. Thereby, the following constraints need to be respected:*

- *The route must start at port 1.*
- *The total turnaround time (which also includes traveling and service time, see* (1)*) must be between $w_{\min}$ and $w_{\max}$ weeks.*
- *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.*
- *Full containers can be transported either in the outbound or inbound direction.*

Let us now more formally define the constraints on the turnaround time, then the profit function. Let $N' = N_{out} \cup N_{in}$. To calculate the total turnaround time in hours, denoted by T_{tot}, 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 at the calling ports, and the fixed time T_{i^*}:

$$T_{tot} = \sum_{i \in N'} \sum_{j \in N'} \left(T_i^l + T_j^u\right) z_{ij} + \sum_{i \in N'} \sum_{j \in N'} \left(\tilde{T}_i^l + \tilde{T}_j^u\right) y_{ij} + \sum_{i \in N_{out}} \left(T_i^a + T_i^d\right) + \sum_{i \in N_{in}} \left(T_i^a + T_i^d\right) + T_{i^*} \qquad (1)$$

The turnaround time in weeks is $w = \lceil \frac{T_{tot}}{168} \rceil$, as there are $7 \times 24 = 168$ hours in a week. The constraint is then

$$w_{min} \leq w \leq w_{max}.$$

The number of idle hours the ship stays immobilized at the initial port is $168 \cdot w - T_{tot} \leq H_I$. Note that, for the given line that is operated on a weekly basis, the value of w implicitly determines the size of the fleet, i.e., for $w = 4$, a fleet of four ships is needed to guarantee the weekly service.

The profit function is then calculated as follows:

$$\sum_{i \in N'} \sum_{j \in N'} P_{ij} z_{ij} - \sum_{i \in N'} \sum_{j \in N'} \left(L_i^f + U_j^f\right) z_{ij} - \sum_{i \in N'} \sum_{j \in N'} \left(L_i^e + U_j^e\right) y_{ij} + \\ - \sum_{i \in N'} \left(L_i l_i + S_i s_i\right) - \sum_{i \in N_{out}} F_i - \sum_{i \in N_{in}} F_i - K_{i^*} - \mathtt{wcc} \cdot w - C_I(168 \cdot w - T_{tot}). \qquad (2)$$

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 (paid only for the called ports). Finally, $\mathtt{wcc} \cdot w$ denotes the *capital cost* (which includes the cost of the charter, maintenance, insurance, crew, etc.) and K_{i^*} denotes the *bunker* (fuel) cost for the whole route from 1 to i^* and back. The value of K_{i^*} is calculated as follows (see [21]):

$$K_{i^*} = \mathtt{P_{out}} \cdot (T_{i^*} - T_{i^*}^w) \cdot (\mathtt{fp} \cdot \mathtt{scf} + \mathtt{lp} \cdot \mathtt{scl}) \qquad i^* \in N. \qquad (3)$$

The value $C_I(168 \cdot w - T_{tot})$ represents the total cost for the ship remaining idle at the initial port.

Observe that the profit calculated by (2) is the profit (in US$) for one ship collected during the turnaround time of w weeks. This is at the same time the *weekly profit* for the shipping company for the *whole fleet*. Indeed, 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 for full and empty containers, respectively, two directed acyclic graphs (DAG) $G^f = (\bar{N}, A^f)$ and $G^e = (\bar{N}, A^e)$, where:

- Each graph has the same ordered set of nodes $\bar{N} = \{1, 2, \dots, n, n+1, n+2, \dots, 2n-1\}$, where 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 $i > n$.

- The shipping of full containers is modeled by defining the set of arcs $A^f = \{(i, j) : i, j \in \bar{N}, i < j \leq n \text{ or } n \leq i < j\}$. The shipping of empty containers (which is unrestricted) is instead modeled using the set of arcs $A^e = \{(i, j) : i \in \bar{N}, j \in \bar{N}, i < j\}$, i.e., contrary to full containers, empty containers can be transported from outbound to inbound while staying on the the ship at the last visited port.

- To each node $i \in \bar{N}$, we associate:
  - $\bar{T}_i$ is the time necessary to visit port i. It is defined as:

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

    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 $\bar{i} = 2n - i$.

- We associate the following parameters to the arcs:
  - $\bar{D}_{ij}$ is the weekly expected demand of full containers between i and j, and it is set as:

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

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

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

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

$$\bar{C}_{ij} := L_i^e + U_j^e, \quad (i, j) \in A^e$$

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

$$\bar{T}_{ij} := \begin{cases} T^l_i + T^u_j, & \text{if } i < j \leq n, \\ T^l_{\bar{i}} + T^u_{\bar{j}}, & \text{if } n \leq i < j \end{cases} \quad (i,j) \in A^f, \qquad \bar{T}^e_{ij} := \tilde{T}^l_i + \tilde{T}^u_j \quad (i,j) \in A^e$$

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 can now be described by selecting a subset of arcs A' and associating with each arc (i, j) in A' a label $z_{ij}|y_{ij}$, which means that z_{ij} full and y_{ij} empty containers are shipped from port i to port j. By construction, only shipping in the outbound, respectively, inbound direction is allowed for full containers. 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. Assume that the time data and w_{max} are such that 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. For the two examples, empty containers do not cross outbound and inbound, so we only represent arcs in A^f. 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.

Another example given in Figure 3 illustrates an alternative solution in which balancing of empty containers ensures that no leasing or storage cost need to be paid.

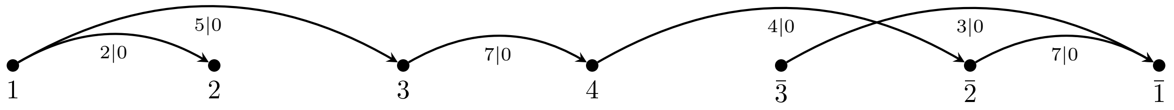


Figure 2: Example of a solution in which storage of empty containers is needed at port 1 (3 containers) and port 4 (3 containers), and leasing is necessary at port 2 (1 container) and port 3 (5 containers).

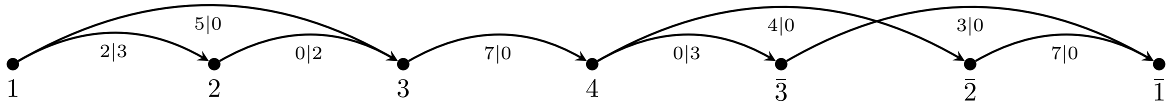


Figure 3: Example of a solution in which no leasing/storage of empty containers is needed, since the balancing is guaranteed by the shipping of empty containers.

3.3. NP-hardness

The BCSP introduced in Definition 1 contains a constraint associated to the upper bound on the total turnaround time. This is a knapsack-type constraint, which implies that in general the BCSP is at least weakly NP-hard. We refer the interested reader to [15, 22] 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 turnaround time constraint is kept, and (2) in the case that the turnaround time 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 $\sum_{(i,j)\in A^f: i\leq i', j>i'} \bar{D}_{ij} \leq C$ for each port $i' \in \bar{N}$. Similarly, we say that the instance is *time-unconstrained*, if the imposed interval $[w_{min}, w_{max}]$ for the turnaround time is sufficiently large so that all ports can be called in both directions and all demands can be served, and $\texttt{wcc} = C_I = 0$ (in this case, decision variables w and h can be removed).

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 instance is:*

- *capacity-unconstrained,*
- *all costs are equal to zero (i.e., $\bar{C}_{ij} = 0$, for all $(i,j) \in A^e$ and $K_i = L_i = S_i = 0$ for all $i \in N$),*
- *all demands and profits are binary (i.e., $\bar{D}_{ij}$, $\bar{P}_{ij} \in \{0,1\}$, for all $(i,j) \in A^f$),*
- *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: $V = \{2, \ldots, n-1\}$. We build the DAG $G^f = (\bar{N}, A^f)$ where $\bar{N} = \{1\} \cup V \cup \{n\} \cup \bigcup_{i=2}^{n-1}\{\bar{i}\} \cup \{2n-1\}$ and $A^f = \bigcup_{i=2}^{n}\{(1,i),(n,\bar{i})\} \cup \bigcup_{i=2}^{n-2}\bigcup_{j=i+1}^{n-1}\{(i,j),(\bar{i},\bar{j})\} \cup \bigcup_{i=2}^{n-1}\{(i,n),(\bar{i},2n-1)\}$

Profits and demands are defined as follows:

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

We set $\bar{T}_i = 168$ (168 hours correspond to exactly one week) for each port $i \in \bar{N}$, $w_{min} = 0$ and $w_{max} = 2k+2$, $T_i = 0$, for $i \in N$, and $\texttt{wcc} = C_I = 0$. Figure 4 illustrates the transformation from the graph H into the DAG G^f, where dashed arcs correspond to the arcs where the associated net profits and demands are equal to zero ($\bar{P}_{24} = \bar{D}_{24} = \bar{P}_{25} = \bar{D}_{25} = \bar{P}_{\bar{5}\bar{2}} = \bar{D}_{\bar{5}\bar{2}} = \bar{P}_{\bar{4}\bar{2}} = \bar{D}_{\bar{4}\bar{2}} = 0$).

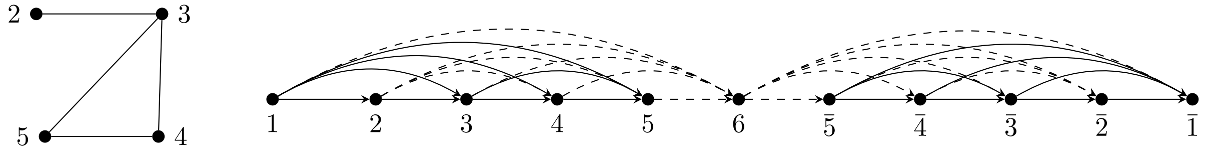


Figure 4: Transformation from CLIQUE problem into the BCSP on the DAG G^f.

Under the assumptions stated in the theorem, we observe that the optimal solution of the BCSP has a value $(k+1)k$ if and only if the selected ports in this solution correspond to a clique of size k in G^f. Indeed, given the interval $[w_{\min}, w_{\max}]$ for the turnaround time in weeks, and given that stopping at any port will exactly spend one week, at most $2k+2$ ports can be called (including the first 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 the last visited port is at most $\frac{(k+1)k}{2}$, and the same holds for the profit collected from the last visited port to 1. If the induced subgraph defined by the visited ports is not complete, then the solution value is strictly less than $(k+1)k$. This also holds if less than k ports are visited between 1 and the last visited port. This completes the proof. □

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 zero (i.e., $L_i = S_i = 0$, for all $i \in N$) and* wcc*=0.*

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 $(i,j) \in A^f$, let binary variables b_{ij} be set to one iff the complete demand $\bar{D}_{ij}$ is satisfied. Assume for the moment that n is the last visited port along the route. 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:

$$\max \sum_{(i,j)\in A^f} \bar{P}_{ij}\bar{D}_{ij}b_{ij} - \sum_{i\in\bar{N}} F_i x_i \tag{4}$$

$$b_{ij} \leq x_i \qquad (i,j) \in A^f \tag{5}$$

$$b_{ij} \leq x_j \qquad (i,j) \in A^f \tag{6}$$

$$x_n = 1 \tag{7}$$

$$x_i, b_{ij} \in \{0,1\} \qquad (i,j) \in A^f \tag{8}$$

The 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.

To deal with the more general case in which the final port has to be chosen, one has to solve the above problem $n-1$ times, assuming the port i, $2 \leq i \leq n$, is chosen as the last one. This concludes the proof. □

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 [21].

The following variables are common in both our models:

- χ_i are binary variables which are set to one iff port i is the last visited port, $i \in N$,

- 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, $(i,j) \in A^f$,

- 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, $i \in N$,
- w is the turnaround time in weeks (which is also the size of the fleet)
- h is the number of idle hours the ship remains immobilized at the starting port.

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

$$A_{i'}^f = \{(i,j) \in A^f : i \leq i', j > i'\} \tag{9}$$

with the corresponding notation $A_{i'}^e$ for empty containers. By summing up the number of all containers shipped through $A_{i'}^f$ and $A_{i'}^e$, 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 the same arc variables y_{ij} as those of the problem definition:

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

and to ease the reading, we use the simplified compact notations :

$$\begin{aligned}
\sum_j (z_{ij} + y_{ij}) \quad &\equiv \quad \sum_{(i,j)\in\delta_f^+(i)} z_{ij} + \sum_{(i,j)\in\delta_e^+(i)} y_{ij} \\
\sum_j (z_{ji} + y_{ji}) \quad &\equiv \quad \sum_{(j,i)\in\delta_f^-(i)} z_{ji} + \sum_{(j,i)\in\delta_e^-(i)} y_{ji}
\end{aligned}$$

In the following we present a first MIP model that will be denoted by M_1^S. It is a valid formulation for the BCSP (notation S stands for *splittable demand*). The objective function is given in (10) and it 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. The last three terms correspond to the fuel cost, the charter cost and the cost for idle hours. Since the cost of visiting the last port of the line should be paid just once and in the set $\bar{N}$ each node except node n is represented twice, the objective function includes the necessary correction term (fourth term in the formula).

$$\max \sum_{(i,j)\in A^f} \bar{P}_{ij} z_{ij} - \sum_{(i,j)\in A^e} \bar{C}_{ij} y_{ij} - \sum_{i\in\bar{N}} F_i x_i + \sum_{i\in N\setminus\{n\}} F_i \chi_i - \sum_{i\in N}(S_i s_i + L_i l_i) - \sum_{i\in N} K_i \chi_i - \texttt{wcc}\ w - C_I h \tag{10}$$

In the following we present the families of constraints of M_1^S. Constraint (11) ensures that exactly one port is chosen as the last visited port, whereas constraints (12) and (13) ensure that this port is called (both in the outbound and inbound direction). Constraints (14) ensure that all ports after the last visited one cannot be called in each direction.

$$\sum_{i\in N} \chi_i = 1 \tag{11}$$

$$\chi_i \le x_i \qquad i \in N \tag{12}$$

$$\chi_i \le x_{2n-i} \qquad i \in N \tag{13}$$

$$\chi_i + x_j \le 1 \qquad i \in N \setminus \{n\}, i < j < 2n - i. \tag{14}$$

Constraints (15) and (16) 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 $\bar{D}_{ij}$.

$$z_{ij} \le \bar{D}_{ij} x_i \qquad (i,j) \in A^f \tag{15}$$

$$z_{ij} \le \bar{D}_{ij} x_j \qquad (i,j) \in A^f \tag{16}$$

Constraints (17) and (18) 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.

$$\sum_j (z_{ij} + y_{ij}) \le C x_i \qquad i \in \bar{N} \tag{17}$$

$$\sum_j (z_{ji} + y_{ji}) \le C x_i \qquad i \in \bar{N} \tag{18}$$

Inequalities (19) 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.

$$\sum_{(i,j)\in A^f_{i'}} z_{ij} + \sum_{(i,j)\in A^e_{i'}} y_{ij} \le C \qquad i' \in \bar{N} \setminus \{\bar{1}\} \tag{19}$$

Balancing of empty containers is given by constraints (20), 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 either $l_i \geq 0$ and $s_i = 0$, or $s_i \geq 0$ and $l_i = 0$, but not $l_i > 0$ and $s_i > 0$.

$$\sum_j (z_{ij} + y_{ij}) - \sum_j (z_{ji} + y_{ji}) + \sum_j (z_{\bar{i}j} + y_{\bar{i}j}) - \sum_j (z_{j\bar{i}} + y_{j\bar{i}}) = l_i - s_i \qquad i \in N \tag{20}$$

Finally, we compute the turnaround time w and the number of idle hours h with constraint (21) together with the minimization of associated costs in the objective. The bounds on variables w and h are given in constraints (22)-(23).

$$168w - h \leq \sum_{i \in N} T_i \chi_i + \sum_{(i,j) \in A^f} \bar{T}_{ij} z_{ij} + \sum_{(i,j) \in A^e} \bar{T}^e_{ij} y_{ij} + \sum_{i \in \bar{N}} \bar{T}_i x_i - \sum_{i \in N \setminus \{n\}} \bar{T}_i \chi_i \leq 168w \tag{21}$$

$$w_{min} \leq w \leq w_{max} \tag{22}$$

$$h \leq H_I \tag{23}$$

The nature of decision variables is defined by (24)-(26). 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.

$$s_i, l_i \geq 0 \qquad i \in N \tag{24}$$

$$x_i \in \{0, 1\} \qquad i \in \bar{N} \tag{25}$$

$$w, h \in \mathbb{Z}_+ \tag{26}$$

$$z_{ij} \in \mathbb{Z}_+ \qquad (i,j) \in A^f \tag{27}$$

$$y_{ij} \in \mathbb{Z}_+ \qquad (i,j) \in A^e \tag{28}$$

We stress that the given MIP formulation is, to the best of our knowledge, the first model in the literature that integrates the optimization of the turnaround time along with the design of the line. On the contrary, most of the available models for the barge container routing explicitly impose the overall turnaround time, which may lead to suboptimal solutions and reduced profits. In our computational study (cf. Section 5) we analyze benefits of our new integrated approach and demonstrate the gains in profit that can be achieved by utilizing our new model.

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

In this section we propose an alternative, more compact but less intuitive formulation for the BCSP. In deriving this new model, we exploit the fact that empty containers can be seen as a *single commodity* that can be picked up and/or delivered at any port. This is in contrast to the full containers, where each of them has a pre-specified origin and destination with a differenciated profit (and hence, each $\bar{D}_{ij}$ has to be considered as a separate commodity). As shown in the following subsection, the new model retains the same quality of the LP relaxation bound, while significantly reducing the number of decision variables. This property can turn into an important advantage when difficult instances with unsplittable demand need to be solved (see our computational results in Section 5 for further details). In this second model (denoted by M_2^S in the following) we do not 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. For the ease of exposition and without loss of generality, the model is derived from the model M_1^S (with explicitly imposed bounds on the turnaround time), 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, $i \in \bar{N}$.

using the transformation:

$$y_i^{in} = \sum_{(j,i)\in\delta_e^-(i)} y_{ji} \tag{29}$$

$$y_i^{out} = \sum_{(i,j)\in\delta_e^+(i)} y_{ij} \tag{30}$$

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:

$$\begin{aligned}
\max \sum_{(i,j)\in A} \bar{P}_{ij} z_{ij} - \sum_{i\in\bar{N}} (F_i x_i + L_i^e y_i^{out} + U_i^e y_i^{in}) + &\\
+ \sum_{i\in N\setminus\{n\}} F_i \chi_i - \sum_{i\in N} K_i \chi_i - \texttt{wcc}\ w - C_I h & \qquad (31)
\end{aligned}$$

$$(11)\text{-}(16)$$

$$\sum_{(i,j)\in\delta_f^+(i)} z_{ij} + y_i^{out} \le C x_i \qquad i \in \bar{N} \tag{32}$$

$$\sum_{(j,i)\in\delta_f^-(i)} z_{ji} + y_i^{in} \le C x_i \qquad i \in \bar{N} \tag{33}$$

$$\sum_{(i,j)\in A_{i'}^f} z_{ij} + \sum_{i\le i'} (y_i^{out} - y_i^{in}) \le C \qquad i' \in \bar{N} \setminus \{\bar{1}\} \tag{34}$$

$$\begin{aligned}
\sum_{(i,j)\in\delta_f^+(i)} z_{ij} + y_i^{out} - \sum_{(j,i)\in\delta_f^-(i)} z_{ji} - y_i^{in} + &\\
+ \sum_{(\bar{i},j)\in\delta_f^+(\bar{i})} z_{\bar{i}j} + y_{\bar{i}}^{out} - \sum_{(j,\bar{i})\in\delta_f^-(\bar{i})} z_{j\bar{i}} - y_{\bar{i}}^{in} = l_i - s_i & \qquad i \in N \qquad (35)
\end{aligned}$$

$$\begin{aligned}
168w - h \le \sum_{i\in N} T_i \chi_i + \sum_{(i,j)\in A^f} \bar{T}_{ij} z_{ij} + &\\
+ \sum_{i\in\bar{N}} (\bar{T}_i x_i + \tilde{T}_i^l y_i^{out} + \tilde{T}_i^u y_i^{in}) - \sum_{i\in N\setminus\{n\}} \bar{T}_i \chi_i \le 168w & \qquad (36)
\end{aligned}$$

$$\sum_{j<i} y_j^{out} - \sum_{j<i} y_j^{in} \ge y_i^{in} \qquad i \in \bar{N} \tag{37}$$

$$\sum_{j>i} y_j^{in} - \sum_{j>i} y_j^{out} \ge y_i^{out} \qquad i \in \bar{N} \tag{38}$$

$$\sum_{i\in\bar{N}} y_i^{out} = \sum_{i\in\bar{N}} y_i^{in} \tag{39}$$

$$y_i^{in}, y_i^{out} \in \mathbb{Z}_+ \qquad i \in \bar{N} \tag{40}$$

$$(22) - (27)$$

Constraints (32)-(36) are the direct adaptation of inequalities (15)-(21), respectively, using the transformation (29)-(30) . In order to balance the empty containers, additional constraints are needed. Constraints (37) 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 inequalities (38) is similar, but it concerns the empty containers loaded at port i. Finally, we impose by (39) that the total number of empty containers loaded all over the route is unloaded, i.e., the ship cargo has no more empty containers at the end of the route. Note that for the last port i^* verifying $\chi_{i^*} = 1$, we necessarily have $y_{i^*}^{out} = y_{i^*}^{in} = 0$ or $y_{\bar{i^*}}^{out} = y_{\bar{i^*}}^{in} = 0$, as since all other ports after i^* are not called and there is no demand between i^* and $\bar{i^*}$, then loading $a > 0$

containers at i^* and unloading $b < a$ containers at $\bar{i^*}$ is strictly more costly than just loading $a - b$ containers at i^* (or $\bar{i^*}$) which keeps the same number of empty containers when the ship leaves the port. The same reasoning holds for unloading at i^* then loading at $\bar{i^*}$. 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 $(\bar{x}, \bar{\chi}, \bar{z}, \bar{s}, \bar{l}, \bar{w}, \bar{h}, y_{ij})$ of model M_1^S can be transformed into a (fractional) solution $(\bar{x}, \bar{\chi}, \bar{z}, \bar{s}, \bar{l}, \bar{w}, \bar{h}, y^{in}, y^{out})$ of model M_2^S with the same objective value, and vice-versa. The linear transformation is given by (29)-(30) on both sides, i.e., also from model M_2^S to M_1^S.*

Proof. The transformation from solutions of M_2^S to M_1^S, that also ensures (29) and (30), will be explained at the end of the proof.

Observe first that if (29) and (30) hold, we have equality of objective values for the two models, which follows from the definition of $\bar{C}_{ij}$, since

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

whereas all the other terms remain equal in the objective functions of the two models. Now, let us focus on constraints. Observe that if (29) and (30) are satisfied, then obviously constraints (17), (18), (20) and (21) for M_1^S become constraints (32), (33), (35) and (36) for M_2^S and vice-versa. Moreover, the cut capacity constraints (19) become constraints (34) and vice versa because

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

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 $i \le i'$ 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 (37) and (38) are satisfied. Indeed, by using (29) and (30), 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, (37) is satisfied. Similarly, the validity of (38) 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 (29) and (30) are satisfied, i.e., this system of equations has a solution.

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^{in} and y^{out}. Recall that the circulation problem consists of finding a network flow with added constraints of a lower bound on edge flows and flow conservation constraints also being required for the source and sink [1]. Our extended graph is constructed starting from the original digraph G^e, 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 $(i, j) \in A^e$, we add an arc (i^+, j^-) with capacity $C - \bar{z}_{ij}$. In Figure 5 we show an example of the extended digraph for an instance with 4 ports (we skip the arcs crossing outbound and inbound to make it more readable). Solving the system of equations (29)-(30) 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 [16], section 7.7) which is exactly property (39). So, we can indeed find the y_{ij} satisfying (29)-(30) from the y_i^{in}, y_i^{out} values. This completes the proof. □

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 5.

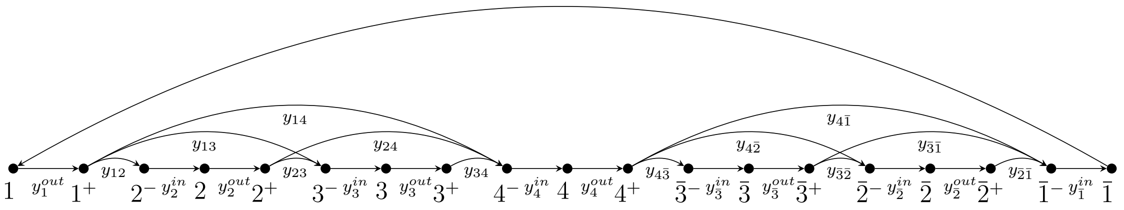


Figure 5: Example of extended graph for an instance of 4 ports.

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 (35) can be rewritten as

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

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 $S_i s_i + L_i l_i + L_i^e(y_i^{out} + y_{\bar{i}}^{out}) + U_i^e(y_i^{in} + y_{\bar{i}}^{in})$ in the objective function (31). 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 ($y^{in} = y^{out} = 0$) will be less costly. Since the empty containers variables y^{in} and y^{out} consume ship capacity in constraints (32), (33), (34) and consume more time in constraint (36) if $T_i^l + T_i^u < \bar{T}_i^l + \bar{T}_i^u$, then these variables y^{in} and y^{out} will all be equal to zero at optimality. □

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. *For the splittable demand case, if (i) there is enough demand to fill the ship at any time (i.e., $\sum_{(i',j)\in A_i^f} \bar{D}_{i'j} \geq C$ 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 $\sum_{(i',j)\in A_i^f} \bar{D}_{i'j} \geq C$ for each port $i \in \bar{N}$ and let $(x^*, z^*, s^*, l^*, y^*)$ be an optimal solution such that there exists a port $i \in \bar{N}$ with $\sum_{(i',j)\in A_i^f} z^*_{i'j} < C$. In this solution, the ship might carry some empty containers (i.e., we might have $\sum_{(i',j)\in A_i^e} y^*_{i'j} \neq 0$) 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 $\sum_{(i',j)\in A_i^f} z'_{i'j} = C$ 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. □

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 possible upper limit for the turnaround time imposed by the profit maximization which implicitly bounds the amount of full containers to be shipped. Similarly, if the demand is not allowed to be split, there will be 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 and storage costs. Both observations are verified in our numerical experiments, as we will see later (cf. Section 5).

4.5. Modeling Unsplittable Demand

In many realistic cases 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". In this case we call the models M_1^U and M_2^U, where U stands for unsplittable.

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 and M_2^S require a slight modification, which consists in replacing z_{ij} by $\bar{D}_{ij} b_{ij}$, where the binary variable b_{ij} is set to one iff the complete demand $\bar{D}_{ij}$ is shipped from i to j, i.e.:

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

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 models M_1^S and M_2^S by $\bar{D}_{ij} b_{ij}$, for all $(i,j) \in A^f$.

Observation 4.1. *The value of the LP-relaxation of model M_1^U (resp., M_2^U) is the same as the one obtained by model M_1^S (resp., M_2^S).*

This result follows because, for model M_1^S, every fractional feasible solution $(\bar{x}, \bar{z}, \bar{s}, \bar{l}, \bar{w}, \bar{h}, \bar{y})$ can be transformed into a fractional feasible solution $(\bar{x}, \bar{z}, \bar{s}, \bar{l}, \bar{w}, \bar{h}, \bar{b})$ for M_1^U with the same objective function value, and vice-versa using the linear transformation $\bar{b}_{ij} = \frac{\bar{z}_{ij}}{D_{ij}}$ for all $(i, j) \in A^f$. 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. The same transformation holds for models M_2^S, which has the same LP-relxation value as M_1^U by Theorem 3, and M_2^U. Accordingly, the gap between the optimal integer solutions value of model M_1^U (resp., M_2^U) and its LP-relaxation value cannot be smaller than the LP-gap of model M_1^S (resp., M_2^S). In our computational study (cf. Section 5), we computationally show that these gaps are considerably higher and accordingly, models M_1^U and M_2^U are harder to solve than their splittable versions. 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. Computational Experiments

The goals of our computational study are as follows: (1) Evaluate the performance of the two formulations introduced in this paper; (2) Compare them with the state-of-the-art model from [21] for a specific subclass of the problem; (3) Demonstrate the economical advantages of our models to simultaneously optimize the route and the turnaround times, rather than imposing the turnaround time explicitly; (4) Measure how the empty container rebalancing is influenced by imposing splittable vs unsplittable demands, both in terms of the cost and the solution time.

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.

5.1. Benchmark Instances

We use the benchmark instances for the BCSP introduced in [21]: 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 [9], where ports along the river Danube are taken as input.

Table 2: The characteristics of 5 different container barge ships (taken from [21]).

	Container barge ships	No. units	TEU	$\mathtt{P}_{\text{out}}$ [kW]	Total TEU	$\mathtt{v}_1$ [km/h]	$\mathtt{v}_2$ [km/h]	`wcc/7` [$]
Ship 1	Motorized cargo push vessel	1	90	2×607	215	19	11	3050
	Pushed barges	1	165	–				
Ship 2	Motorized cargo push vessel	1	145	2×1024	409	20	12	4450
	Pushed barges	2	132	–				
Ship 3	Motorized cargo push vessel	1	77	2×565	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	2×927	338	20	12	3850
	Pushed barges	4	60	–				

Other parameters are taken from [17, 18, 26, 27, 28, 31]. In Table 2 we also report the speed of the ship in the outbound and inbound direction, $\mathtt{v}_1$ and $\mathtt{v}_2$, respectively, and daily charter costs (`wcc/7`) . The Benchmark instances, along with the LP files and the best obtained solutions of all the models presented in this paper are publicly available at `www.mi.sanu.ac.rs/~tanjad/ships.htm`.

5.2. Computational Performance and Comparison with the State-of-the-Art

To the best of our knowledge, no integrated approaches for the BCSP are considered in the previous literature. We therefore compare it with a problem variant that was introduced in [21] in which the turnaround time was fixed ($T_{\max}$), the final port (n) was given and the transport of full and empty containers was allowed only in one direction. The instances considered by [21] have a fixed $T_{\max}$ which depends on the number of ports, i.e., for $n = 10$ the value of $T_{\max}$ is 3 while for all the others $T_{\max}$ is 4. In this section we compare the computational performance of our new models against the state-of-the-art from [21]. Later, in Section 5.3 we show the economical advantages of an integrated approach in which the turnaround time is optimized within a model. We conclude our computational study in Section 5.4 by comparing the performance of the two new models for the more challenging problem variant in which demands are not allowed to be split.

We first consider the BCSP problem variant from [21] and compare the following three settings:

- M_1^S: the MIP formulation introduced in Section 4.1, based on arc-variables y_{ij} for modeling empty-container repositioning, where $\chi_n = 1$, and $\sum_{(i,j)\in A^e\setminus A^f} y_{ij} = 0$ are imposed and $168w$ is replaced by $T_{\max}$ in (21),
- M_2^S: the MIP formulation introduced in Section 4.2, based on node-variables y_i^{in} and y_i^{out} for modeling

empty-container repositioning, where $\chi_n = 1$, $\sum_{i\in N\setminus\{1\}} y_i^{in} = \sum_{i\in N\setminus\{n\}} y_i^{out}$ and $\sum_{i\in\{n,\ldots,2n-2\}} y_i^{out} = \sum_{i\in\{n+1,\ldots,2n-1\}} y_i^{in}$ are imposed and $168w$ is replaced by $T_{\max}$ in (36), and

- MLDM: the MIP formulation studied in [21].

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:

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

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

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

where UB is the global upper bound obtained upon the termination of the algorithm. 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 “profit” 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 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

Table 3: Number of variables and constraints of the different models on the BCSP variant from [21].

	MLDM		Model M_1^S		Model M_2^S	
instance	# 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

time limit after exploring hundreds of thousands of 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.

Comparing the performance of M_1^S vs M_2^S, we can see that M_2^S slightly outperforms M_1^S1 for computation times, although no clear picture emerges: we may conclude that the two models are competitive, both in terms of computing time and number of enumerated branch-and-bound nodes. We will see later that the difference between the two models is much more significant for the case of unsplittable demands.

5.3. Optimizing Turnaround Time

We now focus on the more general case in which we let the optimization model decide about the optimal turnaround time and the size of the fleet, along with the ports to be visited and amounts of containers to be shipped. We compare solutions of the BCSP variant from [21] (with the fixed turnaround time) versus the optimal solutions of the BCSP as introduced in this paper. On 10 out of 20 instances from our benchmark set, we show that the profits can be significantly improved due to the newly proposed integrated modeling approach. These instances are listed in Table 5 along with the basic solution properties including: 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 (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 turnaround time in weeks.

In all reported cases, the optimal solutions are obtained by *decreasing* the given 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. The improvement of profit (which is reported in column Impr.[%]) ranges between roughly 10 and 100%.

Table 4: Computational performance of the three models on the BCSP variant from [21].

		MLDM				Model M_1^S		Model M_2^S		
instance	profit	time [s]	# nodes	lp gap [%]	exit gap	time [s]	# nodes	time [s]	# nodes	lp gap [%]
Port10_1	22339.01	2.43	3699	26.03	0.00	**0.03**	30	**0.03**	30	0.20
Port10_2	24738.23	0.15	249	1.83	0.00	0.11	464	**0.09**	334	0.81
Port10_3	23294.75	4.88	5821	31.88	0.00	**0.04**	84	**0.04**	63	0.14
Port10_4	20686.27	0.42	591	8.76	0.00	0.13	297	**0.09**	108	1.90
Port10_5	25314.99	0.84	1589	13.15	0.00	**0.07**	184	0.09	174	0.81
Port15_1	12268.96	222.01	96914	146.79	0.00	**0.03**	0	**0.03**	0	0.02
Port15_2	25341.81	85.31	54249	69.08	0.00	**0.12**	238	0.15	351	0.05
Port15_3	13798.22	228.95	162075	158.94	0.00	**0.06**	0	**0.06**	13	0.03
Port15_4	22372.58	897.73	523976	180.98	0.00	0.19	432	**0.18**	392	0.32
Port15_5	15799.96	82.42	32528	101.04	0.00	0.08	12	**0.05**	10	0.01
Port20_1	19892.20	TL	842898	136.64	22.95	0.30	90	**0.26**	177	0.26
Port20_2	33204.57	TL	866050	66.81	11.60	0.31	137	**0.20**	99	0.08
Port20_3	21042.63	TL	905052	148.36	23.97	0.86	530	**0.71**	562	0.45
Port20_4	27962.32	TL	1064498	165.02	29.27	0.29	145	**0.26**	126	0.24
Port20_5	24257.89	TL	805063	96.21	16.94	**1.06**	1919	1.10	2041	0.18
Port25_1	21843.22	TL	403415	134.33	37.46	1.28	281	**1.02**	370	0.45
Port25_2	34410.74	TL	363195	66.88	23.07	1.08	332	**0.57**	138	0.25
Port25_3	23443.93	TL	401944	144.84	39.80	6.11	5777	**5.57**	5342	0.45
Port25_4	29177.52	TL	453145	161.98	40.89	1.19	298	**0.80**	181	0.61
Port25_5	26189.80	TL	416339	95.25	28.44	5.19	4272	**4.50**	4157	0.30

Table 5: Solution features for the optimal turnaround case. Only improved solution, when compared to those shown in Table 4 are reported.

	BCSP with $T_{\max} = 4$, [21]						BCSP with optimal turnaround time						
instance	# calls	% D_1	% D_2	Load [%]	fleet	profit	# calls	% D_1	% D_2	Load [%]	fleet	profit	Impr.[%]
Port15_1	27	42.65	48.36	96.1	4	12269.0	14	23.84	75.64	94.3	3	24297.3	98.0
Port15_2	28	53.39	56.69	95.1	4	25341.8	15	28.58	80.97	85.5	3	30199.1	19.2
Port15_3	27	41.72	47.30	96.3	4	13798.2	14	23.71	75.21	94.7	3	25272.9	83.2
Port15_5	28	50.07	53.17	94.9	4	15800.0	14	29.70	85.78	91.4	3	29702.2	88.0
Port20_1	28	28.79	53.97	96.6	4	19892.2	13	15.22	85.66	94.5	3	24667.9	24.0
Port20_3	28	28.68	49.90	97.3	4	21042.6	14	14.57	77.33	94.9	3	25544.5	21.4
Port20_5	29	32.42	55.00	96.5	4	24257.9	14	18.85	85.59	91.5	3	30000.9	23.7
Port25_1	27	21.35	56.23	97.7	4	21843.2	13	10.89	85.93	95.1	3	24829.9	13.7
Port25_3	28	20.76	52.89	97.6	4	23443.9	14	10.29	77.11	96.4	3	25732.8	9.8
Port25_5	28	23.82	63.49	97.3	4	26189.8	14	13.54	85.59	90.5	3	30000.9	14.6

5.4. Unsplittable Demands

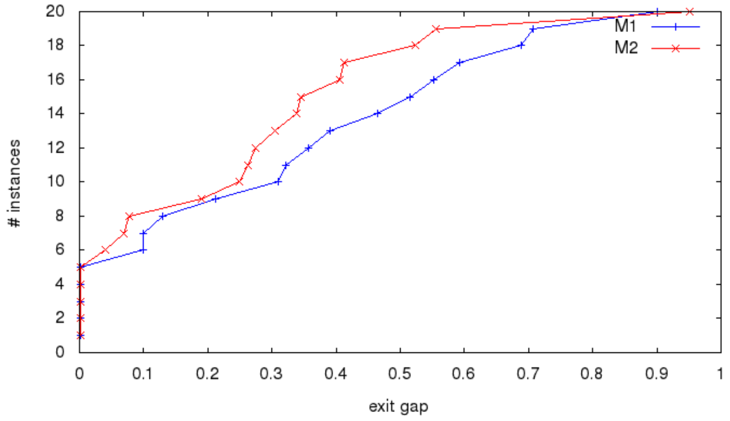
In this section we focus on the BCSP with unsplittable demand and we take a closer look at the performance of the two proposed models, namely M_1^U and M_2^U. Recall that both models exhibit the same quality of lower bounds, but since M_2^U contains significantly less variables, we investigate how beneficial is the use of the latter model in some practical situations.

The same benchmark setting from [21] is used, apart from the fact that the demand is not allowed to be split. Unsplittable demands make the BCSP much harder to solve from the computational perspective. Both models do not manage to solve all the instances of this problem variant within the given time limit of one hour. The obtained exit gaps are shown in Figure 6. A point (x, y) in this chart indicates that for y instances, the exit gap was $\leq x\%$. The figure shows that both models are able to solve only 5 instances to proven optimality within one hour of computing time. For the instances in which the time limit is reached, the chart shows that model M_2^U consistently outperforms M_1^U, i.e., the exit gaps of model M_2^U are globally lower than the ones of M_1^U and they range between 0.1% and 1%.

In a second set of experiments we enlarged the time limit to 30 hours and we ran the experiments in a multi-core setting on the same machine (with 8 CPUs). That allowed us to solve seven more instances to optimality. The results of this comparison are given in Table 6. The following values are provided for all 12 instances solved to proven optimality by at least one of the two models: the optimal solution value (profit), and for each of the two models, the total CPU time in seconds and the number of branch-and-bound nodes.

Table 6 shows that for all instances with up to 15 ports, we are able to provide optimal solution values.

Figure 6: Exit gaps of models M_1^U and M_2^U after one hour of CPU time.



In addition, two more instances with 20 ports (namely Port20_1 and Port20_4) are solved by the model M_2^U within approximately 23.5 and 13.5 hours, whereas the model M_1^U runs into memory limit. For the remaining 8 instances, both models run into memory or time limit. Further increasing of the time limit to 75 hours, does not help in solving more instances to optimality.

The obtained results indicate that the sparser model, namely M_2^U significantly outperforms M_1^U: for the difficult instances with 15 ports, the average speed-up ratio is 2.7, but one of the five instances remains unsolved by M_1^U, due to the memory overconsumption (denoted by ML in Table 6). We therefore conclude that it indeed pays off to project out arc variables and model the empty container repositioning as a single commodity using node-variables only.

5.5. Analyzing Solutions: Splittable vs Unsplittable Demands

In the following, we analyze the profit of BCSP solutions for both splittable and unsplittable demand cases. The 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. The 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.

We summarize the increase of profits that can be achieved by allowing splittable demands and/or empty container repositioning in Figure 7. The zero line corresponds to the *basic setting* in which the demand cannot be split and no empty container repositioning is allowed. We then demonstrate: (1) the relative increase of profit (in %) if empty container repositioning is allowed (curve denoted by "+e"), and (2) the 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

Table 6: Computational performance of models M_1^U and M_2^U for unsplittable demands. The time limit was set to 30h.

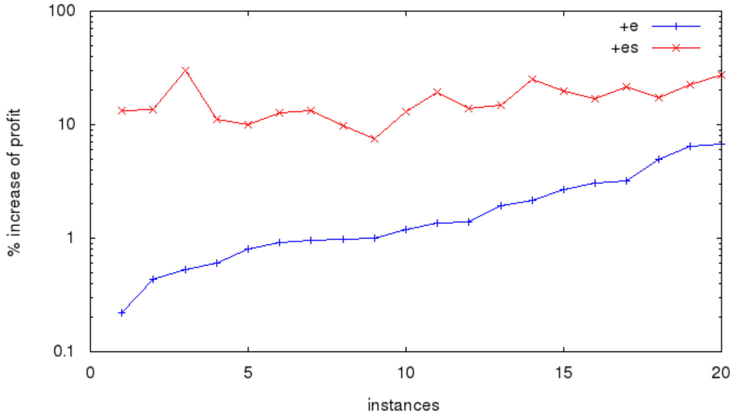
		Model M_1^U		Model M_2^U		
instance	profit	time [s]	# nodes	time [s]	# nodes	lp gap [%]
Port10_1	18204.52	**4.02**	4097	11.25	8320	4.10
Port10_2	21872.31	**0.85**	778	1.19	1669	2.71
Port10_3	19783.36	**3.25**	2377	7.95	6401	3.59
Port10_4	17608.09	**0.37**	0	1.10	206	6.37
Port10_5	23297.05	**1.47**	1820	2.27	4253	2.28
Port15_1	9658.81	ML	ML	**91001.05**	95207142	2.21
Port15_2	22706.26	29912.24	6672144	**7138.04**	10436315	1.48
Port15_3	11574.65	19424.34	3507591	**15584.75**	15253603	1.99
Port15_4	19913.30	28201.77	4546113	**24435.47**	23252439	3.23
Port15_5	13561.16	41305.43	7434705	**10147.12**	14549527	1.45
Port20_1	17270.88	ML	ML	**95431.86**	86459157	2.34
Port20_4	25138.57	ML	ML	**48863.55**	33264924	3.39

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%.

A further comparison shows that the profit, in the case of splittable demands, is higher from 8% to 22 % compared to the unsplittable 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 by large transport demands.

The results obtained 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 ([21, 36]). 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. The results for instances with 10 ports, characterized by smaller total demands,

Figure 7: Relative improvement of profit, compared to the setting in which loads cannot be split and no empty container repositioning is allowed (zero line).



are in line with this claim since the highest profit is not achieved for the ship with the largest TEU capacity. For instances with 15, 20 and 25 ports, barge container ship 2, with the highest carrying capacity, reached the highest values of objective function.

6. Conclusion

In this article we studied the design of a route for a liner shipping company that provides regular service among a sequence of ports on a fixed-schedule basis. The models have been derived from the perspective of the shipping company that maximizes its profit, 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 in which the turnaround time is fixed, we proposed an integrated approach that simultaneously optimizes the route, the cargo and the turnaround time (and hence, the size of the fleet as well).

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 the optimization of turnaround time, better profits (between 10% and 100%) can be achieved, (ii) profits can also be increased by allowing empty container repositioning and demand splitting.

Concerning the future work, it would be interesting to consider non-identical routes performed by a fleet of ships that is not necessarily homogeneous, and incorporation of transshipment in the context of outbound-inbound shipping with empty container repositioning.

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.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993. ISBN 0-13-617549-X.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] B. D. Brouer, C. V. Karsten, and D. Pisinger. Optimization in liner shipping. *4OR*, 15(1):1–35, 2017.

[7] M. Christiansen, K. Fagerholt, and D. Ronen. Ship routing and scheduling: Status and perspectives. *Transportation Science*, 38(1):1–18, 2004.

[8] 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.

[9] COLD. Container Liner Service Danube. Technical report, via Donau, ÖIR, Port of Constantza, Final Report, Vienna, 2006.

[10] 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.

[11] K. Fagerholt, M. Christiansen, L. M. Hvattum, T. A. Johnsen, and T. J. Vabø. A decision support methodology for strategic planning in maritime transportation. *Omega*, 38(6):465 – 474, 2010.

[12] 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`.

[13] 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.

[14] C. V. Karsten. *Competitive Liner Shipping Network Design.* PhD thesis, Technical University of Denmark, 2015.

[15] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems.* Springer, Berlin Heidelberg, 2004.

[16] J. Kleinberg and E. Tardos. *Algorithm design.* Pearson Education India, 2006.

[17] 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.

[18] 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.

[19] E. Malaguti, S. Martello, and A. Santini. The traveling salesman problem with pickups, deliveries, and draft limits. *Omega*, 2017. doi: https://doi.org/10.1016/j.omega.2017.01.005.

[20] 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.

[21] 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.

[22] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations.* John Wiley & Sons, Chichester, New York, 1990.

[23] 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.

[24] 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.

[25] E. K. Norlund, I. Gribkovskaia, and G. Laporte. Supply vessel planning under cost, environment and robustness considerations. *Omega*, 57(Part B):271 – 281, 2015.

[26] 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.

[27] Port of Antwerp, Sept 30, 2017. URL `www.portofantwerp.com`.

[28] Z. Radmilović, B. Dragović, and V. Maraš. Power unit – cargo space link in transport. *YUJOR*, 15(1): 50–58, 2005.

[29] 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.

[30] K. Rana and R. Vickson. Routing container ships using lagrangean relaxation and decomposition. *Transportation Science*, 25(3):201–214, 1991.

[31] Rotterdam Port Information, Sept 30, 2017. URL `www.rotterdamportinfo.com`.

[32] 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.

[33] 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.

[34] 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.

[35] D.-P. Song and J.-X. Dong. Empty container repositioning. In *Handbook of Ocean Container Transport Logistics*, pages 163–208. Springer, 2015.

[36] M. Stopford. *Maritime Economics*. Routledge, Oxford, UK, 2009.

[37] UNCTAD. Uniated nations conference on travel and development: Review of maritime transport 2015, 2015. URL `http://unctad.org/en/pages/PublicationWebflyer.aspx?publicationid=1374`.

[38] X. Wang, K. Fagerholt, and S. W. Wallace. Planning for charters: A stochastic maritime fleet composition and deployment problem. *Omega*, 2017 (forthcoming).

[39] 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 8 to 12 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 the furthest port in the upstream direction.

Figure 8: Optimal route of barge container ship for 10 possibly calling ports and schematic overview of obtained container flows. Instance Port10_1.

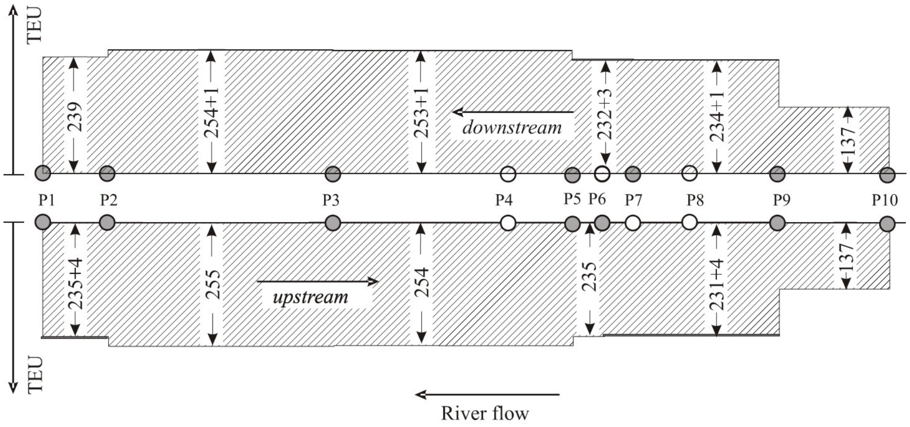


Figure 9: Optimal route of barge container ship for 10 possibly calling ports and schematic overview of obtained container flows. Instance Port10_2.

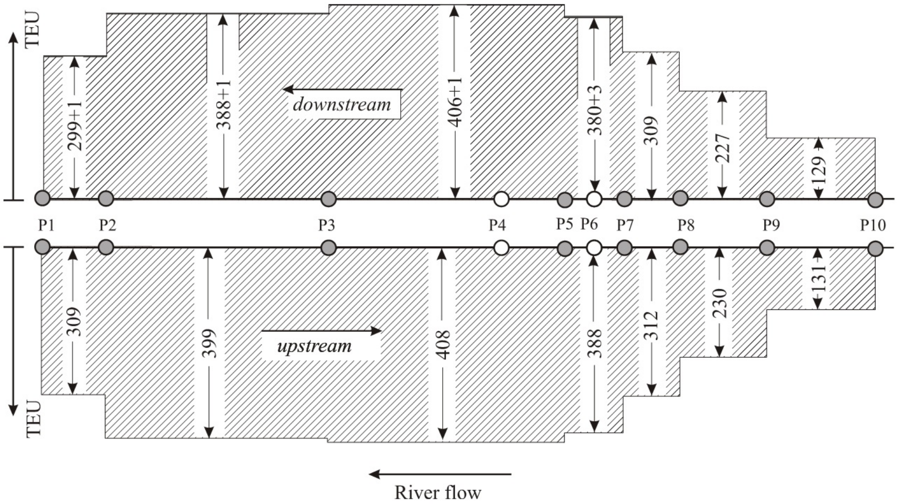


Figure 10: Optimal route of barge container ship for 10 possibly calling ports and schematic overview of obtained container flows. Instance Port10_3.

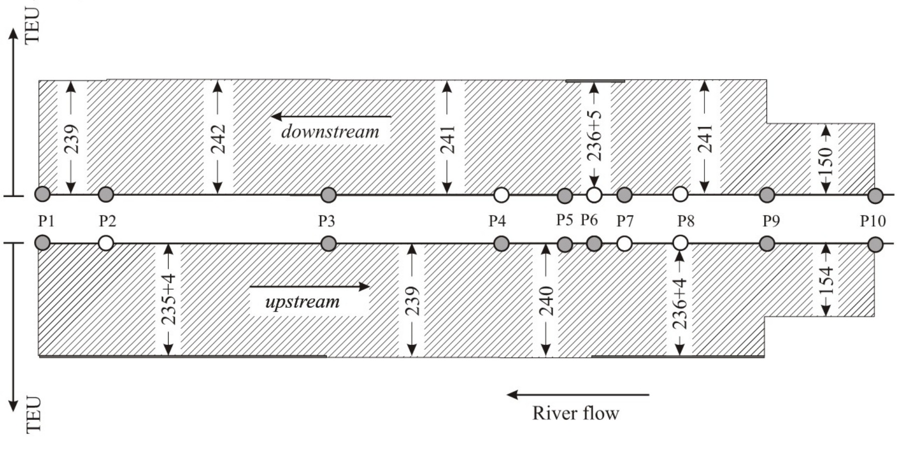


Figure 11: Optimal route of barge container ship for 10 possibly calling ports and schematic overview of obtained container flows. Instance Port10_4.

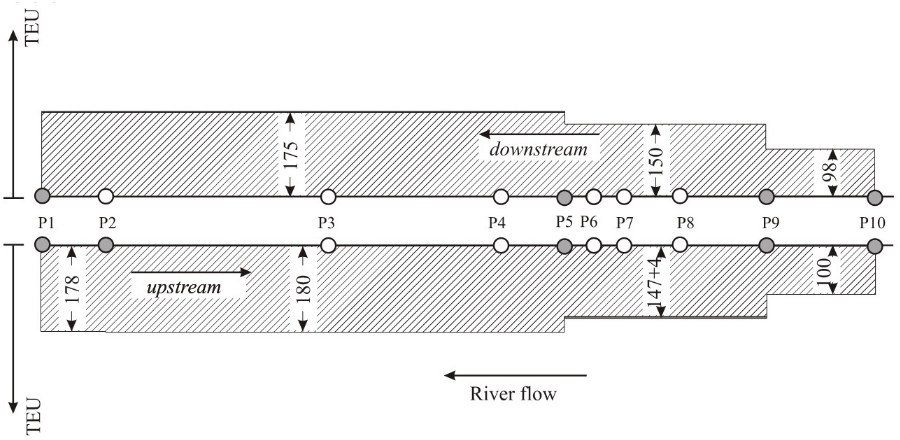
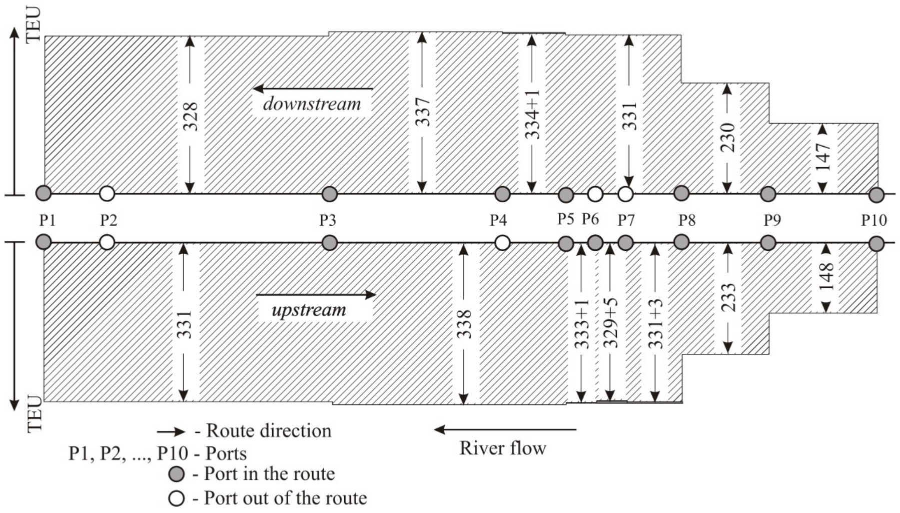


Figure 12: Optimal route of barge container ship for 10 possibly calling ports and schematic overview of obtained container flows. Instance Port10_5.


TEU
328
downstream
337
334+1
331
230
147
P1
P2
P3
P4
P5
P6
P7
P8
P9
P10
331
upstream
338
333+1
329+5
331+3
233
148
TEU
→ - Route direction
P1, P2, ..., P10 - Ports
● - Port in the route
○ - Port out of the route
River flow