Intl. Trans. in Op. Res. xx (2020) 1–17

A new formulation for the liner shipping network design problem

Marie Ameln[a], Julie Sand Fuglum[a], Kristian Thun[a], Henrik Andersson[a,*], Magnus Stålhane[a]

[a] *Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Gløshaugen, Alfred Getz vei 3, 7491 Trondheim, Norway*

Received DD MMMM YYYY ; received in revised form DD MMMM YYYY ; accepted DD MMMM YYYY

Abstract

The liner shipping network design problem is an important problem within liner shipping since a good network can reduce costs and increase profits. Given sets of ports, vessel classes and demands between the ports, the problem is to design a network of cyclic routes and assign a vessel class to each route so that all demand can flow through the network at minimal cost. In this paper we analyze a new formulation of the liner shipping network design problem based on a two layer network structure. The formulation takes into account the cost of transshipment and allows for complex service structures. Valid inequalities and a novel approach of inner representations of low dimensional polyhedra are proposed. A new set of small instances with up to twelve ports has been developed and the formulation has been tested on these instances. Instances with up to ten ports are solved to optimality, but largest instances are not, confirming that the liner shipping network design problem is a very complex problem. The proposed improvements of the formulation are also shown to have a positive effect.

Keywords: Liner shipping; network design; complex route structures

1. Introduction

Maritime shipping represents the most important source of transportation in international trade, and its share of the total global trade in volume has been estimated to lie around 85% (Drewry, 2014). Basically, seaborne shipping has a monopoly on transporting large quantities of goods between continents. Acquisitions of new vessels represent a huge capital investment, and thus it is crucial for shipping companies to utilize their fleets in the best possible way. Most companies rely on manual planning of fleet schedules, which usually works reasonably well due to experienced planners. However, in the last decades there have been many mergers and acquisitions with the result that fleets have become larger and the scheduling more complex. Thus, optimization based methods might have a great potential for improving fleet routing and scheduling.

There are generally three modes of operations in shipping: industrial, tramp, and liner (Lawrence, 1972). Industrial shipping is characterized by cargo owner also owning the vessels. Tramp vessels are like taxis as they follow the available cargo and liners are operated somewhat like bus lines with a published itinerary and schedule (Christiansen et al., 2004, 2012).

Liner shipping is characterized by cyclic routes repeatedly sailed during a scheduled horizon and transshipment of cargo in hub ports. Liner vessels have huge capacities which makes the transportation very efficient. Liner shipping companies publish their service routes, with fixed

* Corresponding Author : henrik.andersson@ntnu.no

sequences of ports of call at a regular service frequency, to attract cargo. The combination of all services constitute a liner shipping network.

The liner shipping industry has contributed to advances in the standard of living for most of the world's population in the last 35 years, as the gains from trade through advancing global commerce were enabled by a reliable, efficient and relatively low-cost transportation provided by the industry (Insight, 2009). According to the World Shipping Council (WSC) there are approximately 400 liner services in operation today, most providing weekly departures from all the ports that each service calls.

An efficient liner shipping network is of great importance both economically and environmentally. The process of designing the service routes of a liner shipping company is thus essential for the competitiveness of the company and its ability to sustain its share of the global containerized freight market. The problem of determining the structure of the services of the network is often referred to as the liner shipping network design problem (LSNDP). An introduction to liner shipping is given by (Brouer et al., 2014a), where a benchmark suite for the LSNDP, called LINER-LIB-2012, is also presented. Since the seminal paper by Agarwal and Ergun (2008), a rich literature on different aspects of liner shipping network design has arisen. (Agarwal and Ergun, 2008) formulate the problem over a space-time network and incorporate a heterogeneous fleet, a weekly service frequency, multiple vessel routes, and cargo transshipment operations, but transshipment costs are not considered in the network design stage. Multiple visits to the same port are allowed, as long as they happen on different week days. Reinhardt and Pisinger (2011) argue that transshipment of goods is frequently occurring in liner shipping and the associated cost should not be ignored when designing the network. They present a flow-based model and propose a branch-and-cut algorithm to solve it. The formulation allows butterfly routes where at most one port is visited at most twice and are able to solve small instances to optimality. Later, Plum et al. (2014) propose a compact formulation where artificial service nodes are introduced that allows non-simple cycles with any number of calls to one or more ports. The model is solved using commercial software and tested on two feeder networks from LINER-LIB-2012, see Brouer et al. (2014a). In Thun et al. (2017), the effect of allowing complex service structures when designing the network is investigated. The analysis shows that complex service structures can create more cost-efficient networks. Recently, Santini et al. (2018) study the special case of feeder network design without transshipments, where one port serves as origin or destination for all demand. A path-flow formulation and a branch-and-price algorithm is proposed and different versions of two instances from LINER-LIB-2012 are solved.

Besides these exact algorithms, a number of heuristic approaches has been proposed. Alvarez (2009) combine tabu search and column generation into a heuristic that also handles different speed options. The algorithm cannot handle butterfly routes and transshipments in a satisfactory way. (Brouer et al., 2014a) later extend the work of Alvarez (2009) to correctly handle transshipments. The effect of combining fleet composition, vessel scheduling and cargo routing is investigated by Mulder and Dekker (2014). A key element in the proposed matheuristic is the aggregation of ports into cluster, where each cluster is served by a feeder network. Aggregation is also used by Jepsen et al. (2011), where demand is aggregated and defined between regions instead of between ports. A branch-and-price algorithm is proposed, but it did not perform well in practice. Brouer et al. (2014b) develop a matheuristic where integer programming is used to select a set of improving port insertions and removals on each service. A computational study on LINER-LIB 2012 shows promising results. A broader discussion of recent research on strategic, tactical, and operational problems within container shipping is found in Meng et al. (2014), while Tran and Haasis (2015) and Brouer et al. (2017) are two recent survey article focusing on network design in liner shipping and optimization in liner shipping respectively.

This paper presents a new formulation of the LSNDP that allows for most route structures that are seen in today's liner shipping networks. The main contributions of this paper are:

– A flow-based formulation of the liner shipping network design problem that takes into account

the cost of transshipment and allows for complex route structures.
- A novel technique to strengthen the formulation using inner representation of low-dimensional polyhedra.

The rest of the paper is organized as follows. Section 2 contains the problem description together with the mathematical formulation and different ways to strengthen the formulation. Section 3 provides a computational study, and Section 4 gives some concluding remarks.

2. Problem description and mathematical formulation

To describe the liner shipping network design problem (LSNDP), we define a set of ports, a set of demands, a set of vessel classes and a distance matrix with all port to port distances. Each demand is defined by an origin, a destination and a quantity and each vessel class is defined by the number of vessels belonging to the class, the capacity and operational parameters such as speed interval and fuel consumption. The LSNDP is then the problem of designing a set of services that allow all demands to be served. A service is a cyclic route visiting a subset of ports. A demand can be served either by one service visiting both the origin and destination of the demand and providing enough capacity or the demand can be transshipped between services. This means that the demand follows one service from the origin to an intermediate destination and then another service to the destination or another intermediate destination. A demand can be split between different options. Each service is operated by a single vessel class, and multiple vessels are deployed to the service sailing one week apart to allow for a weekly frequency of port calls.

There is no standard mathematical formulation of the LSNDP since each liner service has specific constraints based on strategic decisions (Reinhardt and Pisinger, 2011). This has given rise to different formulations including various assumptions. To allow for complex route structures and transshipments, we propose an arc-flow formulation on a two layer network. The formulation allows for services visiting all ports twice except for two ports where the service can change layers. This enables butterfly routes, see (Reinhardt and Pisinger, 2011), as well as pendulum routes, see Tran and Haasis (2015). The formulation can create more complex route structures than (Reinhardt and Pisinger, 2011) and generalizes the subproblem in Thun et al. (2017) to handle multiple routes and the interaction between them. A drawback with the formulation is that it bounds the number of times each port can be visited by a route, but this can be remedied by introducing more layers. It is therefore slightly less flexible than the formulation in Plum et al. (2014), but there the flexibility comes at the prize of a much more complex formulation. In practice though, more than two visits to the same port are very rare.

Figure 1 shows different route structures that are allowed in the proposed arc-flow formulation. To the left is a simple route where no ports are visited more than once. In the middle a butterfly route is shown where the gray port is visited twice. The right network shows a complex route structure, several ports are visited more than once.

Let P denote the number of ports. We define the problem on a graph $G = (\mathcal{N}, \mathcal{A})$ where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of arcs. The network has two layers and the nodes $\mathcal{N} = \{1, \ldots, 2P\}$ are divided into an upper layer $\mathcal{N}^U = \{1, \ldots, P\}$ and a lower layer $\mathcal{N}^L = \{P + 1, \ldots, 2P\}$.

Each port p is represented by both node p and node $P + p$. The arc set is defined as $\mathcal{A} = \{(i,j)|i,j \in \mathcal{N}^U\} \cup \{(i,j)|i,j \in \mathcal{N}^L\} \cup \{(i, P+i)|i \in \mathcal{N}^U\} \cup \{(P+i, i)|i \in \mathcal{N}^U\}$, i.e. it is only possible to change layers by going from a node in one layer to the node representing the same port in the other layer. Figure 2 shows an example of a route in a two layer network. To the left is the route and a possible seqeunce of the arcs. The dark gray nodes in the right figure are $\mathcal{N}^U$ and the white nodes are $\mathcal{N}^L$. The port that is visited twice does not have a connection between the layers, this makes it impossible to transship without paying the transshipment cost.

There is a set of demands $\mathcal{D}$ and each demand is denoted (o, d) where o is the origin port of the

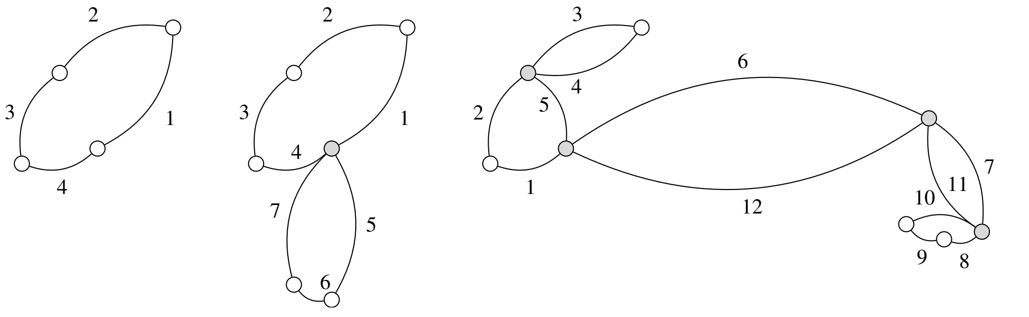

Fig. 1. Three different route structures. To the left is a simple route, in the middle a butterfly route where one port (marked gray) is visited twice on the route. To the right is a complex route structure where several ports are visited twice (marked gray). The number on each arc gives a possible sequence order.

demand and d is the destination port. The weekly demand from port o to port d is denoted D_{od}. If a demand is transshipped at port p, there is a per unit transshipment cost C_p^T associated with this. The fleet available is heterogeneous but divided into a set of vessel classes $\mathcal{K}$. The capacity and number of vessels of class k is denoted Q_k and N_k respectively. The port fee for a vessel of class k visiting port p is C_{pk}^P, and the sailing cost per week and fixed cost of using a vessel of class k is denoted C_k^S and C_k^V respectively. Also, let T_{ijk} denote the sailing time from port i to port j by a vessel of class k. All routes

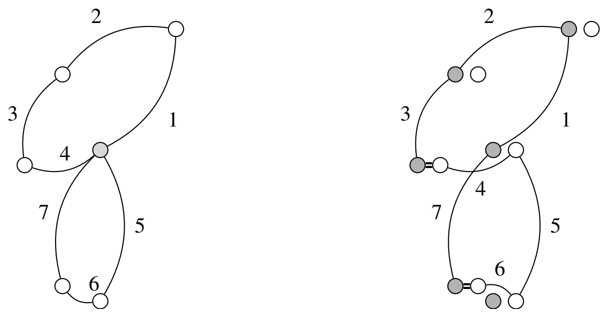

Fig. 2. The two layer network. To the left is a butterfly route where one port (marked gray) is visited twice on the route. To the right is the same route in the two layer network. Each port is duplicated and the network consists of two layers (slightly shifted and marked dark gray and white respectively). Note that the route changes layers between arcs 3 and 4 (marked by a double line) and between arcs 6 and 7.

The model is an arc-flow formulation and thus it is going to construct the routes, i.e. it does not have a set of predefined routes to choose from. Even though neither the number nor the composition of the routes in the optimal solution are known beforehand, the model still needs a way of referring to a specific route. This is resolved by introducing the set of routes $\mathcal{R}_k$ for each vessel class k and a binary variable y_{kr} that is 1 if route r is sailed by a vessel of class k. We also introduce the variable x_{ijkr} which is 1 if a vessel of class k sailing route r is using arc (i, j) and 0 otherwise. The continuous variable f_{ijodkr} represents the flow of the demand (o, d) transported by a vessel of class k on arc (i, j) on route r and t_{iodkr} is the quantity of the demand (o, d) that is transshipped in port i onto a vessel of class k sailing route r. The demands and capacities are all integers, and all flows are therefore naturally integer. The number of vessels of class k sailing route r is denoted w_{kr}. To only allow routes to travel between regions once, there is an upper bound $\overline{T}$ on the duration of each route, which also specify the maximum number of vessels that can be assigned to each route. To enhance readability, the constraints are grouped and each group is presented separately. The arc-flow model of the problem may then be formulated as follows:

$$\min \sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}_k} C_k^V w_{kr} + \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}_k} C_k^S T_{ijk} x_{ijkr} + \\ \sum_{i\in\mathcal{N}^U}\sum_{j\in\mathcal{N}^U}\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}_k} C_{ik}^P (x_{ijkr} + x_{P+i,P+j,kr}) + \\ \sum_{i\in\mathcal{N}^U}\sum_{(o,d)\in\mathcal{D}}\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}_k} C_i^T (t_{iodkr} + t_{P+i,odkr}) \tag{1}$$

The objective function (1) minimizes total cost. The cost function of the LSNDP consists of the fixed cost of using vessels, the sailing costs of the vessels assigned to routes, port fees and transshipment costs.

Route constraints

$$\sum_{i\in\mathcal{N}^U}\sum_{j\in\mathcal{N}^U} 2(x_{ijkr} + x_{P+i,P+j,kr}) \\ + \sum_{i\in\mathcal{N}^U} (x_{i,P+i,kr} + x_{P+i,ikr}) - 6y_{kr} \geq 0 \qquad k\in\mathcal{K}, r\in\mathcal{R}_k \tag{2}$$

$$\sum_{i\in\mathcal{N}} x_{ijkr} - y_{kr} \leq 0 \qquad j\in\mathcal{N}, k\in\mathcal{K}, r\in\mathcal{R}_k \tag{3}$$

$$\sum_{i\in\mathcal{N}} x_{ijkr} - \sum_{i\in\mathcal{N}} x_{jikr} = 0 \qquad j\in\mathcal{N}, k\in\mathcal{K}, r\in\mathcal{R}_k \tag{4}$$

Constraints (2) force y_{kr} to 0 if no arcs for route r are used. The formulation does not allow a route to use both arc (i,j) and (j,i) and the shortest route only using arcs from the upper layer therefore visits three ports, while the route from port i to port j and back again uses one arc from the upper and lower layer respectively and two arcs between the layers. Therefore, the arc weights differ between the different arcs. Each node can only be visited at most once, as stated in constraints (3). The node balances of the routes are handled in constraints (4).

Flow constraints

$$\sum_{i\in\mathcal{N}^U}\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}_k} (f_{ijodkr} + f_{P+i,P+j,odkr} - f_{jiodkr} - f_{P+j,P+i,odkr}) = 0 \\ j\in\mathcal{N}^U, (o,d)\in\mathcal{D}, j\neq o, j\neq d, \tag{5}$$

$$\sum_{j\in\mathcal{N}^U}\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}_k} (f_{ojodkr} + f_{P+o,P+j,odkr}) = D_{od} \\ (o,d)\in\mathcal{D}, \tag{6}$$

$$\sum_{i\in\mathcal{N}^U}\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}_k} (f_{idodkr} + f_{P+i,P+o,odkr}) = D_{od} \\ (o,d)\in\mathcal{D}, \tag{7}$$

$$\sum_{(o,d)\in\mathcal{D}} f_{ijodkr} \leq Q_k x_{ijkr} \\ (i,j)\in\mathcal{A}, k\in\mathcal{K}, r\in\mathcal{R}_k \tag{8}$$

Constraints (5) are demand flow conservation constraints in the transition ports, i.e. the ports that are neither origin or destination of the given demand. Constraints (6)-(7) state that all cargoes are loaded and unloaded in their origin and destination ports, respectively. Constraints (8) ensure that the vessel capacity is always a bound on the total flow of demand on an arc.

Transshipment constraints

$$\sum_{j\in\mathcal{N}}(f_{ijodkr}-f_{jiodkr})\le t_{iodkr} \qquad i\in\mathcal{N},(o,d)\in\mathcal{D},i\neq o,i\neq d,k\in\mathcal{K},r\in\mathcal{R}_k \tag{9}$$

Constraints (9) set the value of the transshipment variable in the transition ports for demand (o,d) on route r. The transshipment is thus associated with the route that picks up the cargo at the port.

Fleet constraints

$$w_{kr}\ge\sum_{i\in\mathcal{N}^U}\sum_{j\in\mathcal{N}^U}T_{ijk}(x_{ijkr}+x_{P+i,P+j,kr}) \qquad k\in\mathcal{K},r\in\mathcal{R}_k \tag{10}$$

$$\sum_{r\in\mathcal{R}_k}w_{kr}\le N_k \qquad k\in\mathcal{K} \tag{11}$$

Constraints (10) calculate the number of vessels of vessel class k that are needed on route r and constraints (11) make sure that the number of vessels used does not exceed the number of available vessels for each vessel class.

Subtours, i.e. two or more disjoint routes having the same route index, is a problem in the formulation. To avoid subtours we introduce a variable s_{ikr} representing the sequence number of node i in route r of vessel class k counted from an artificial depot and a variable d_{ikr} which is 1 if port i is the artificial depot of route r of vessel class k and 0 otherwise. By imposing that the sequence numbering starts at the artificial depot and must increase along a route, we ensure that there are no subtours. This resemble the Miller-Tucker-Zemlin (MTZ) constraints, (Miller et al., 1960), for the traveling salesman problem, but with the strengthening proposed by Desrochers and Laporte (1991) and a modification for the depot.

Subtour elimination constraints

$$\sum_{i\in\mathcal{N}}d_{ikr}=y_{kr} \qquad k\in\mathcal{K},r\in\mathcal{R}_k \tag{12}$$

$$\begin{aligned}s_{ikr}-s_{jkr}&+(|\mathcal{N}|-1)x_{ijkr}\\&+(|\mathcal{N}|-3)x_{jikr}\le|\mathcal{N}|-2+|\mathcal{N}|d_{jkr} \qquad i\in\mathcal{N},j\in\mathcal{N},k\in\mathcal{K},r\in\mathcal{R}_k\end{aligned} \tag{13}$$

Constraints (12) force each route that is used to have exactly one artificial depot. The depot serves as a reference point for the ordering of all the nodes in a route to ensure that they are connected. Constraints (13) give each node in a route a number according to its order in the sequence starting from the depot. This numbering is essential to avoid subtours, since it ensures that all nodes must be connected to the depot through the variable s_{ikr}, and since there is only one depot in each route it will not be possible with more than one route for each route index.

Variable restrictions

$$x_{ijkr}\in\{0,1\} \qquad (i,j)\in\mathcal{A},k\in\mathcal{K},r\in\mathcal{R}_k \tag{14}$$

$$y_{kr}\in\{0,1\} \qquad k\in\mathcal{K},r\in\mathcal{R}_k \tag{15}$$

$$d_{ikr}\in\{0,1\} \qquad i\in\mathcal{N},k\in\mathcal{K},r\in\mathcal{R}_k \tag{16}$$

$$w_{kr} \in \{0, 1, \ldots, \overline{T}\} \qquad k \in \mathcal{K}, r \in \mathcal{R}_k \tag{17}$$

$$f_{ijodkr} \geq 0 \qquad (i,j) \in \mathcal{A}, (o,d) \in \mathcal{D}, k \in \mathcal{K}, r \in \mathcal{R}_k \tag{18}$$

$$t_{iodkr} \geq 0 \qquad i \in \mathcal{N}, (o,d) \in \mathcal{D}, k \in \mathcal{K}, r \in \mathcal{R}_k \tag{19}$$

$$s_{ikr} \geq 0 \qquad i \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R}_k \tag{20}$$

2.1. Calculating the number of routes for each vessel class

The maximum number of routes that can be used by each vessel class k, i.e. $|\mathcal{R}_k|$, must be decided a priori. This can be done by setting $|\mathcal{R}_k|$ to a large number for each k. Since the size of the problem in terms of constraints and variables is highly correlated with $|\mathcal{R}_k|$ we have developed a heuristic to calculate the maximum number of routes needed.

To estimate the upper bound on the number of routes we use one heuristic for hub-and-spoke networks and one for the feeder networks. A feeder network typically has one port, called the hub, where all demand either has its origin or destination and other ports in the region surrounding it, see Figure 3. Hubs and their surrounding regions are then connected through interregional routes in a hub-and-spoke network, see Figure 4.

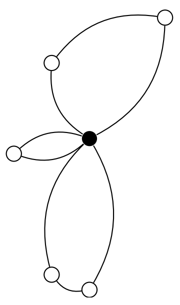
Fig. 3. Feeder network with three routes, the black node is the hub.

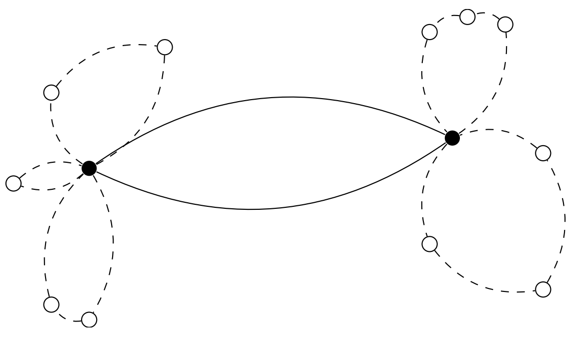
Fig. 4. Hub-and-spoke network with two regions, one hub in each region, one interregional route (solid line), three feeder routes in the left region (dashed lines) and two feeder routes in the right (dashed lines).

In the feeder cases we use a heuristic inspired by the sweep heuristic for vehicle routing problems, see for example Laporte et al. (2000). The ports are sorted according to the angle between the line from the port to the hub and the horizontal line. A route starts at the hub and then visits the other ports in order until the total demand from the hub to the visited ports or the total demand from the visited ports to the hub is higher than the capacity of the vessel class. The route then returns to the hub and a new route starts. When all ports are included in a route, the number of routes generated is divided by two and rounded up. This is then the maximum number of routes for that vessel class. The heuristic is repeated for all vessel classes.

An assumption used by the heuristic for the hub-and-spoke cases is that only the largest vessel class is used for interregional travels. Because of economies of scale, this is a valid assumption in most cases. Given this, the number of routes for the largest vessel class is calculated as the maximum sum of interregional demands divided by the vessel class capacity rounded up. For the smaller vessel classes, the number of routes in each region is estimated and the sum is the maximum number of routes for the vessel class. The estimate is based on transport work, i.e. quantity on board times distance. First the intraregional transportation work is calculated, i.e. demand between two ports in the same region times the distance. Then, the minimum transport work needed if one port in the region is the hub and all interregional demand is handled through that port is calculated. The transportation work a vessel can perform is given by its capacity and speed, and the number of vessels needed to do all transportation work defines the maximum number of routes for the vessel class within the region. The sum for the regions then gives the maximum number of routes for the vessel class.

2.2. *Strengthening the formulation*

Symmetry breaking constraints

Symmetry is a common problem in many combinatorial problems. Symmetrical solutions are different mathematical solutions that represent the same practical solutions. One way of dealing with symmetry is to add constraints that eliminate symmetric solutions (Margot, 2010). These constraints cut away feasible solutions but guarantee to keep at least one symmetric optimal feasible solution.

The symmetry induced by the two layer network is reduced by the following constraints:

$$\sum_{i \in \mathcal{N}^U} d_{ikr} = y_{kr} \qquad k \in \mathcal{K}, r \in \mathcal{R}_k \tag{21}$$

$$\sum_{i \in \mathcal{N}^U} \sum_{j \in \mathcal{N}^U} x_{ijkr} - \sum_{i \in \mathcal{N}^L} \sum_{j \in \mathcal{N}^L} x_{ijkr} \geq 0 \qquad k \in \mathcal{K}, r \in \mathcal{R}_k \tag{22}$$

$$d_{ikr} + x_{i,P+i,kr} \leq 1 \qquad i \in \mathcal{N}^U, k \in \mathcal{K}, r \in \mathcal{R}_k \tag{23}$$

Constraints (21) state that the artificial depot must be a node in the upper layer and replace (12) together with $d_{ikr} = 0, i \in \mathcal{N}^L, k \in \mathcal{K}, r \in \mathcal{R}_k$. The number of arcs in the upper layer must be at least as high as the number of arcs in the lower layer, this is handled in (22). We force all routes to use an arc in the upper layer when leaving the depot in constraints (23).

The sequence numbers s_{ikr} used in the subtour elimination constraints (12) and (13) cause symmetry, and constraints (24) are therefore added to force the artificial depot of a route to have route index 0.

$$(|\mathcal{N}| - 1)(1 - d_{ikr}) \geq s_{ikr} \qquad i \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R}_k \tag{24}$$

The set of routes for each vessel class k, $\mathcal{R}_k$, also induces symmetry. To reduce this symmetry, constraints (25) and (26) are added to force the time duration and utilization of the first route to be at least as high as the second and so forth.

$$\sum_{(i,j) \in \mathcal{A}} T_{ijk}(x_{ijkr} - x_{ijk,r-1}) \leq 0 \qquad k \in \mathcal{K}, r \in \mathcal{R}_k, r > 1 \tag{25}$$

$$y_{kr} - y_{k,r-1} \leq 0 \qquad k \in \mathcal{K}, r \in \mathcal{R}_k, r > 1 \tag{26}$$

The choice of depot is arbitrary given the subtour elimination constraints (12), (13), and (24). By adding

$$\sum_{i \in \mathcal{N}} i \cdot d_{ikr} \leq |\mathcal{N}| - (|\mathcal{N}| - j) \sum_{l \in \mathcal{N}} x_{jlkr} \qquad j \in \mathcal{N}, k \in \mathcal{K} r \in \mathcal{R}_k \tag{27}$$

the visited node with the lowest index is the depot.

Valid inequalities

The transportation work inequalities stated in (28) with F_{ij} being the distance between nodes i and j force the transportation work done by the fleet to be at least as high as the minimum transportation work needed to transport all demands directly from origin to destination.

$$\sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}_k} Q_k F_{ij} x_{ijkr} \geq \sum_{(o,d)\in\mathcal{D}} D_{od} F_{od} \tag{28}$$

Valid capacity inequalities including flows between subsets of ports are given by constraints (29). The set of nodes is divided into two disjoint subsets $\mathcal{N}^S$ and $\overline{\mathcal{N}}^S$ in such a way that the nodes representing the same ports are never in different subsets. Corresponding sets of arcs $\mathcal{A}^S = \{(i,j)|i \in \mathcal{N}^S, j \in \overline{\mathcal{N}}^S\} \cup \{(P+i, P+j)|i \in \mathcal{N}^S, j \in \overline{\mathcal{N}}^S\}$ and $\overline{\mathcal{A}}^S = \{(i,j)|(j,i) \in \mathcal{A}^S\}$ are created where $\mathcal{A}^S$ are all arcs starting in $\mathcal{N}^S$ and ending in $\overline{\mathcal{N}}^S$ while $\overline{\mathcal{A}}^S$ are all arcs starting in $\overline{\mathcal{N}}^S$ and ending in $\mathcal{N}^S$. The flow from $\mathcal{N}^S$ to $\overline{\mathcal{N}}^S$ is then used as a lower bound on the capacity assigned to routes using arcs in $\mathcal{A}^S$. The right side of (29) is the demand with origin in $\mathcal{N}^S$ and destination in $\overline{\mathcal{N}}^S$, the flow of the same demand leaving $\overline{\mathcal{N}}^S$ and the extra flow that leaves the subset. This flow is the flow leaving $\mathcal{N}^S$ with destination in $\mathcal{N}^S$ and the flow leaving $\mathcal{N}^S$ with both origin and destination in $\overline{\mathcal{N}}^S$. The set of demands fulfilling this is denoted $\mathcal{D}^S$, i.e. $\mathcal{D}^S = \{(o,d) \in \mathcal{D}|d \in \mathcal{N}^S \vee (o \in \overline{\mathcal{N}}^S \wedge d \in \overline{\mathcal{N}}^S)\}$

$$\begin{aligned}&\sum_{(i,j)\in\mathcal{A}^S}\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}_k} Q_k x_{ijkr} \geq \sum_{o\in\mathcal{N}^S}\sum_{d\in\overline{\mathcal{N}}^S} D_{od} + \\ &\sum_{(i,j)\in\overline{\mathcal{A}}^S}\sum_{o\in\mathcal{N}^S}\sum_{d\in\overline{\mathcal{N}}^S}\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}_k} f_{ijodkr} + \sum_{(i,j)\in\mathcal{A}^S}\sum_{(o,d)\in\mathcal{D}^S}\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}_k} f_{ijodkr} \qquad \mathcal{N}^S \subset \mathcal{N}\end{aligned} \tag{29}$$

A version of the valid capacity inequalities without the flow variables are introduced and strengthened using the l-Gomory procedure, see Wolsey (1998). These constraints are shown in (30). By changing α, different valid inequalities can be generated.

$$\sum_{(i,j)\in\mathcal{A}^S}\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}_k} \left\lceil \frac{Q_k}{\alpha} \right\rceil x_{ijkr} \geq \left\lceil \sum_{o\in\mathcal{N}^S}\sum_{d\in\overline{\mathcal{N}}^S} \frac{D_{od}}{\alpha} \right\rceil \qquad \mathcal{N}^S \subset \mathcal{N} \tag{30}$$

Disaggregated capacity constraints
The connection between the route and flow variable expressed in (8) can be strengthened by instead using the disaggregated capacity constraints in (31).

$$f_{ijodkr} \leq \min\{D_{od}, Q_k\} x_{ijkr} \qquad (i,j) \in \mathcal{A}, (o,d) \in \mathcal{D}, k \in \mathcal{K}, r \in \mathcal{R}_k \tag{31}$$

Instead of substituting (8) with (31), disaggregated capacity constraints directly from the origin or directly to the destination can be added to the formulation. This is stated in constraints (32) and (35).

$$\begin{aligned} f_{ojodkr} &\leq \min\{D_{od}, Q_k\} x_{ojkr} && (o,j) \in \mathcal{A}, (o,d) \in \mathcal{D}, k \in \mathcal{K}, r \in \mathcal{R}_k && (32)\\ f_{P+o,jodkr} &\leq \min\{D_{od}, Q_k\} x_{P+o,jkr} && (P+o,j) \in \mathcal{A}, (o,d) \in \mathcal{D}, k \in \mathcal{K}, r \in \mathcal{R}_k && (33)\\ f_{idodkr} &\leq \min\{D_{od}, Q_k\} x_{idkr} && (i,d) \in \mathcal{A}, (o,d) \in \mathcal{D}, k \in \mathcal{K}, r \in \mathcal{R}_k && (34)\\ f_{i,P+d,odkr} &\leq \min\{D_{od}, Q_k\} x_{i,P+d,kr} && (i,P+d) \in \mathcal{A}, (o,d) \in \mathcal{D}, k \in \mathcal{K}, r \in \mathcal{R}_k && (35)\end{aligned}$$

Strengthening the subtour elimination constraints
The following constraints are added to put an upper bound on the sequence variables and to force the sequence number to zero if a node is not visited

$$s_{ikr} \leq \sum_{(l,j)\in\mathcal{A}} x_{ljkr} \qquad i \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R}_k \tag{36}$$

$$s_{ikr} \leq (|\mathcal{N}|-1) \sum_{j\in\mathcal{N}} x_{ijkr} \qquad i \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R}_k \tag{37}$$

Inner representation of low dimensional polyhedra
Instead of deriving valid inequalities from capacity constraints using the 1-Gomory procedure as in (30), the convex hull of the feasible region of each capacity constraint can be described using an inner representation. Using the sets $\mathcal{N}^S$, $\overline{\mathcal{N}}^S$ and $\mathcal{A}^S$ and introducing an integer variable $\chi_k^S = \sum_{(i,j)\in\mathcal{A}^S} \sum_{r\in\mathcal{R}_k} x_{ijkr}$ denoting the number of times a vessel of class k leaves $\mathcal{N}^S$, valid capacity inequalities can be written as

$$\sum_{k\in\mathcal{K}} Q_k \chi_k^S \geq \sum_{o\in\mathcal{N}^S} \sum_{d\in\overline{\mathcal{N}}^S} D_{od} \qquad \mathcal{N}^S \subset \mathcal{N} \tag{38}$$

If the number of vessel classes, $|\mathcal{K}|$, is small, the set of feasible extreme points of the convex hull of the sets defined by constraints (38) intersected with $\mathcal{X} = \{\chi_k^S \in \mathbb{Z}_{\geq}, k \in \mathcal{K}\}$ can easily be generated. For the given set associated with $\mathcal{N}^S$, denote its extreme points $\chi^{(p)}, p = 1, \ldots, \Lambda^S$ where Λ^S is the number of such extreme points. Given this, the valid inequalities (30) associated with $\mathcal{N}^S$ can be replaced by the inner representation

$$\chi_k^S = \sum_{p=1}^{\Lambda^S} \chi^{(p)} \lambda_p^S \qquad \sum_{p=1}^{\Lambda^S} \lambda_p^S \geq 1 \qquad \lambda_p^S \geq 0, p = 1, \ldots, \Lambda^S \tag{39}$$

To illustrate this, consider the case where we have two vessel classes with $Q_1 = 16$ and $Q_2 = 9$ respectively. For a given partition of the nodes into $\mathcal{N}^S$ and $\overline{\mathcal{N}}^S$, assume that $\sum_{o\in\mathcal{N}^S} \sum_{d\in\overline{\mathcal{N}}^S} D_{od} = 26$. Plugging this into (30) gives

$$\left\lceil \frac{16}{\alpha} \right\rceil \chi_1^S + \left\lceil \frac{9}{\alpha} \right\rceil \chi_2^S \geq \left\lceil \frac{26}{\alpha} \right\rceil \tag{40}$$

The total number of valid inequalities that can be generated for $\alpha \geq 1$ is 49, starting from $(16, 9, 26)$ and going to $(1, 1, 1)$ decreasing one number one unit each time, but since 2 is a common divisor of 26 and 16, only 48 of these are generated. There are two copies, which means that there are in total 46 different valid inequalities. Only two of these are non-dominated

$$5\chi_1^S + 3\chi_2^S \geq 9 \tag{41}$$

$$\chi_1^S + \chi_2^S \geq 2 \tag{42}$$

Figure 5 captures the difference between the 1-Gomory procedure and the inner representation. The dashed line corresponds to the original capacity constraint (38), the solid lines are the two non-dominated valid inequalities (41) and (42), and the dotted line marks the boundary of the convex hull. The yellow area is cut away by (41) and (42), and the green area is further cut away by the inner representation (39).

The convex hull can be represented by the constraints

$$3\chi_1^S + 2\chi_2^S \geq 6, \qquad \chi_1^S \geq 0, \qquad \chi_2^S \geq 0 \tag{43}$$

The set of extreme points in this case is $\Lambda^S = \{(2,0),(0,3)\}$ and we can thus replace (30) with

$$\chi_1^S = 2\lambda_1^S + 0\lambda_2^S \qquad \chi_2^S = 0\lambda_1^S + 3\lambda_2^S \qquad \sum_{p=1}^{\Lambda^S} \lambda_p^S \geq 1 \qquad \lambda_p^S \geq 0, p = 1, \ldots, \Lambda^S \tag{44}$$

The feasible solution $\chi_1^S = 1$, $\chi_2^S = 2$ corresponds to $\lambda_1^S = 0.5$ and $\lambda_2^S = 2/3$.

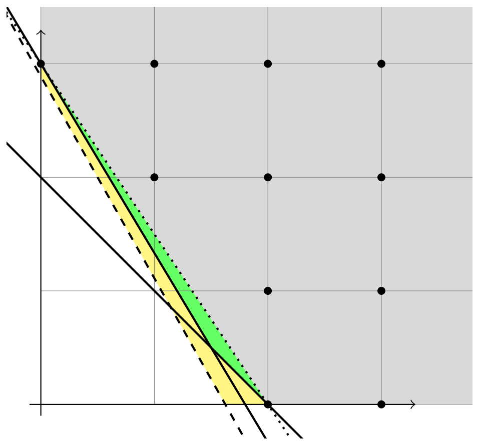

Fig. 5. The feasible region of a capacity set. The dashed line defines the convex hull of the capacity inequalities (38) together with non-negativity constraints. The yellow area is cut away if the non-dominated valid inequalities (41) and (42) (solid lines) generated by the 1-Gomory procedure are added. Using the inner representation (44), the green area is also cut away. The dotted line marks the boundary of the convex hull when the inner representation is used. The gray area is the convex hull of the integer set defined by (38)

3. Computational study

This section presents the results from the computational study performed to test the formulation presented in the paper. The test instances used are presented in Section 3.1. The results from preliminary testing where different settings are tested are given in Section 3.2. The best settings are then used on a larger set of instances and the results from these tests are presented in Section 3.3. All models were implemented in Xpress ver 8.0.4 with default settings. The computational study was performed on a computer with an Intel E5-2670v3 - 12 core, 2 $\times$ 2.3 GHz processor with 64 GB of RAM.

3.1. Test instances

The instances used in this computational study are randomly created to reflect two network structures, feeder networks and hub-and-spoke networks. In the feeder network cases, one port, called the hub, is centered and the other ports are generated in a circular sector around it, see Figure 6. All demands either originates or are destined for the hub. The hub-and-spoke networks are generated by having two regions with ports, see Figure 7. Most of the demands are interregional but some are also between ports in the same region. The number of ports in each region is the same.

To find appropriate distributions, information about demands, vessels, and costs are gathered from the LINER-LIB-2012 benchmark suite, Brouer et al. (2014a). We have extracted data from the Baltic and WorldLarge instances to get appropriate distributions for demand sizes, port fees,

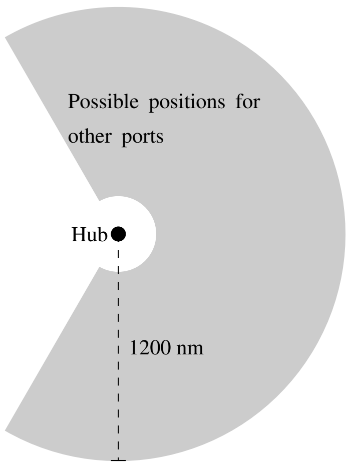

Fig. 6. Feeder network

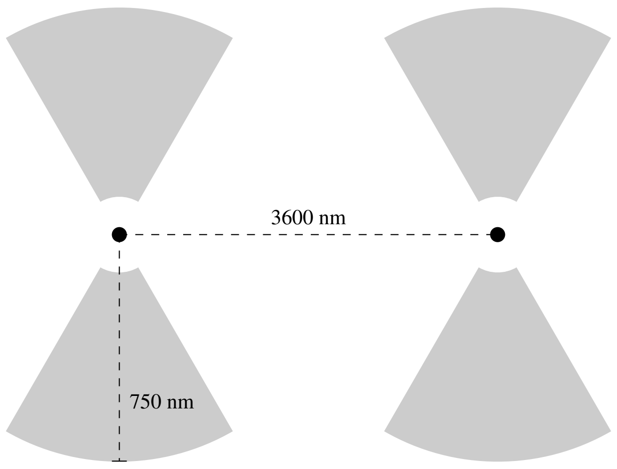

Fig. 7. Hub-and-spoke network

and transshipment costs. These distributions are then used to generate the instances used in this study. We use two vessel classes in the feeder instances and three in the hub-and-spoke instances. General data about the instances are presented in Table 1. The two numbers for average demand size for the hub-and-spoke network reflect intra-regional and inter-regional demands respectively.

Table 1
General data about the instances

Characteristics	Feeder	Hub-and-spoke
Avg. demand size	250	250 / 1000
# vessel classes	2	3
Fixed vessel cost	35' / 56'	35' / 56' / 147'
Capacity	450 / 800	450 / 800 / 2400
Speed (nm/h)	12 / 14	12 / 14 / 16
Consumption (tons/day)	18.8 / 23.7	18.8 / 23.7 / 57.4
Avg. port fee	26' / 27'	26' / 27' / 69'
Avg. transshipment cost	130	130

Each generated instance is characterized by the structure of the network, the number of ports, and the number of demands. The instances are named S_P_D, where $S = F$ means a feeder network and $S = H$ means a hub-and-spoke network. P is the number of ports and D is the number of demands. Note that in a feeder network, $D = 2(P-1)$ since all demands either originates or are destined for the hub. This means that the instance H_6_10 is a hub-and-spoke network with six ports and ten demands.

3.2. Preliminary testing

We have selected a subset of all instances for the preliminary testing. The goal is to analyze the performance of the different strategies to strengthen the formulation in order to come up with a preferable setting for the computational study. We have chosen two indicators to evaluate the settings, the optimal value of the linear relaxation and the lower bound found after one hour of running time. Studying the optimal value of the linear relaxation is useful when comparing different valid inequalities, but it is not useful when assessing the symmetry breaking constraints since these usually do not affect the optimal value of the linear relaxation. The reason for studying the lower bound after one hour is twofold. First, adding valid inequalities to strengthen the formulation improves the root node bound but may also slow down the solution of each node in the branch-and-bound tree. Analyzing the the lower bound after one hour can give insight into the total effect of the inequalities. Second, the gap depends on both the upper and lower bounds and the upper bound is in many cases found by a heuristic. Comparing the lower bounds is therefore clearer.

In all instances tested we allow up to four routes for each vessel class and each route has a maximum duration of four weeks. The total number of vessels of each class is not binding. The settings tested are presented in Table 2. When testing the valid inequalities (29), (30) and (38), the maximum size of $\mathcal{N}^S$ is four, i.e. $|\mathcal{N}^S| \leq 4$.

Table 2
An overview of the tested settings

Name	Constraints included
Basic	(2)-(11), (13)-(20), (21)-(24)
SB:I	Basic + (25)-(26)
SB:II	Basic + (27)
VI:I	Basic + (28)
VI:II	Basic + (29)
VI:III	Basic + (30)
VI:IV	Basic + (31)
VI:V	Basic + (32)-(35)
VI:VI	Basic + (36)-(37)
VI:VII	Basic + (38)-(39)

The results from these preliminary tests are shown in Table 3. For each of the instances tested, the relative gaps between the linear relaxation and lower bound after one hour and the best solution found are presented. The average gaps are then presented in the last two columns. The relative gap between the linear relaxation and the best solution is defined as $LP = 100 \cdot (\overline{z}_{IP} - z_{LP})/(\overline{z}_{IP} - z_{LP}^{BASIC})$, where z_{LP} is the objective value of the linear relaxation, z_{LP}^{BASIC} is the objective value of linear relaxation of the original formulation, and $\overline{z}_{IP}$ is the best objective value found for all settings presented in Table 3. Likewise, the relative gap between the lower bound after one hour and the best solution found is defined as $LB = 100 \cdot (\overline{z}_{IP} - z_{LB})/(\overline{z}_{IP} - z_{LP}^{BASIC})$, where z_{LB} is the lower bound after one hour.

Table 3
Results from testing the initial settings

	F_4_6		F_8_14		F_12_22		H_4_6		H_6_8		H_8_10		Avg	
Setting	LP	LB	LP	LB	LP	LB	LP	LB	LP	LB	LP	LB	LP	LB
Basic	100	0	100	4	100	31	100	0	100	0	100	29	100	10
SB:I	100	0	100	0	100	30	100	0	100	0	100	29	100	10
SB:II	100	0	100	0	100	30	100	0	100	0	100	34	100	11
VI:I	100	0	100	0	100	31	100	0	100	0	100	27	100	10
VI:II	55	0	100	0	100	31	100	0	100	0	100	31	92	10
VI:III	28	0	30	3	41	32	76	0	63	0	72	50	52	14
VI:IV	53	0	26	18	28	28	66	0	57	0	54	50	48	16
VI:V	53	0	26	20	29	28	66	0	57	0	54	46	48	16
VI:VI	100	0	100	20	100	30	100	0	100	0	100	32	100	14
VI:VII	28	0	30	0	38	29	44	0	35	0	52	28	38	9

The symmetry breaking constraints do not affect the lower bound after one hour much, but for the instances that are solved to optimality SB:I and SB:II only use 25% and 36% of the solution time of Basic, respectively. It is clear that the inner representation of small polyhedra, VI:VII, really strengthens the formulation and also produces good lower bounds after one hour. Based on the results in Table 3, a large set of combinations of symmetry breaking constraints and valid inequalities have also been tested. Table 4 shows the combinations that scores best with respect to the criteria: the highest sum of objective values from the linear relaxation, the highest sum of objective values from the lower bounds after one hour, the shortest total computational time, and the fewest number of nodes for the instances solved to optimality. For the criteria highest sum of objective values from the linear relaxation, several combinations gains the same sum. The common denominator of these are that they include the disaggregated capacity constraints VI:IV and the inner representation of low dimensional polyhedra VI:VII; the other valid inequalities did not affect the sum, this is marked with '-'.

Table 4
Best combinations according to the four criteria

Criteria	SB:I	SB:II	VI:I	VI:II	VI:III	VI:IV	VI:V	VI:VI	VI:VII
LP	-	-	-	-	-	1	-	-	1
LB	1	0	0	0	1	0	1	0	1
Time	1	0	0	0	0	0	0	0	1
Nodes	1	0	0	0	0	0	0	0	1

Considering the results presented in Table 4 and the sum of the rankings for each criteria for the different combinations, some conclusions can be drawn. SB:I and VI:VII are included in all the best scoring and has a sum of rankings of $10 + 2 + 1 + 1 = 14$, i.e. this combination has the 10^{th} highest sum of objective values from the linear relaxation, the second highest sum of objective values from the lower bounds after one hour, the shortest total computational time, and the fewest number of nodes for the instances solved to optimality. The second best combination has a sum of rankings of 24, and the combination SB:I and VI:VII is therefore chosen.

3.3. Computational results

The best setting found in the preliminary testing have been used to solve all instances generated. The computational time is set to ten hours. In all instances tested we allow up to four routes for each vessel class and each route has a maximum duration of four weeks. The total number of vessels of each class is not binding.

The results from these tests are presented in Table 5. The table shows the name of the instance, the root node gap calculated as $LP = 100 \cdot (\overline{z}_{IP} - z_{LP})/\overline{z}_{IP}$, the final gap calculated as $LB = 100 \cdot (\overline{z}_{IP} - z_{LB})/\overline{z}_{IP}$ and the computational time. Instances that are solved to proven optimality are marked with a 0 in the LB column. Columns 5 and 6 show the number of routes used and the number of transshipments performed. The number of routes used is presented with the smallest vessel class to the left.

The smaller instances are all solved to proven optimality, but it is clear from Table 5 that the larger instances are very hard to solve. The number of routes used is low compared with $|\mathcal{R}_k|$. We have tested the heuristic described in Section 2.1 to see how the formulation performs based on the maximum number of routes allowed.

The results from using the heuristics together with the best settings are presented in Table 6. The table shows the name of the instance, the root node gap, the final gap and the computational time. The gaps are calculated as in Table 5. Columns 5 - 7 show the maximum number of routes allowed for each vessel class calculated by the heuristic, the number of routes used, and the number of transshipments performed. The number of routes used and the maximum number of routes allowed are presented with the smallest vessel class to the left. Instances that are solved to proven optimality are marked with a 0 in the LB column.

Table 6 shows a clear improvement in the results. Two more instances are solved to proven optimality and the total solution time of the instances that are solved to proven optimality has decreased by 27%. It is also clear that the heuristic used to calculate the maximum number of routes for each vessel class mostly works as intended. For all instances solved to proven optimality without the use of the heuristic, the heuristic produces a maximum number of routes that is equal to or higher than the number of routes in the optimal solution. Since the heuristic calculates the maximum number of routes for each vessel class individually, it overestimates the number which

Table 5
Results on all instances using the best setting

Inst.	LP	LB	Time	#Routes	#Ts
F_4_6	12.8	0	3	0/1	0
F_6_10	5.2	0	14	0/1	0
F_8_14	14.4	0	544	0/2	0
F_10_18	14.5	5.4	36000	1/3	0
F_12_22	23.3	17.4	36000	2/2	0
H_4_4	11.2	0	4	2/1/0	0
H_4_6	13.6	0	8	1/0/2	1
H_4_8	13.5	0	9	0/2/1	0
H_6_6	12.5	0	84	1/1/1	0
H_6_8	11.4	0	89	1/2/1	1
H_6_10	12.0	0	61	1/1/1	1
H_6_12	16.0	0	3340	1/1/2	1
H_6_14	20.8	0	2500	0/0/2	2
H_6_16	14.9	0	2923	0/2/2	1
H_8_8	21.3	0	13141	0/2/1	0
H_8_10	18.5	0	20874	1/2/1	0
H_8_12	23.0	8.7	36000	1/0/2	2
H_8_14	17.4	2.0	36000	0/1/2	3
H_8_16	23.2	13.8	36000	1/0/3	5
H_10_10	18.8	4.5	36000	0/1/2	3
H_10_12	30.0	19.8	36000	0/1/2	3
H_10_14	23.7	13.9	36000	0/2/2	5
H_10_16	26.9	19.1	36000	0/1/3	4

Table 6
Results on all instances using the best setting

Inst.	LP	LB	Time	Heur $\|\mathcal{R}_k\|$	#Routes	#Ts
F_4_6	12.8	0	1	2/1	0/1	0
F_6_10	5.2	0	3	2/1	0/1	0
F_8_14	14.4	0	1006	3/2	0/2	0
F_10_18	14.4	4.6	36000	3/2	1/2	0
F_12_22	26.2	21.6	36000	4/2	2/2	0
H_4_4	11.2	0	1	2/2/1	2/1/0	0
H_4_6	13.6	0	3	2/2/2	1/0/2	1
H_4_8	13.5	0	4	2/2/2	0/2/1	0
H_6_6	12.5	0	43	2/2/2	1/1/1	0
H_6_8	11.4	0	62	3/2/2	1/2/1	1
H_6_10	12.0	0	22	2/2/2	1/1/1	1
H_6_12	16.0	0	2801	3/2/3	1/1/2	1
H_6_14	20.8	0	1237	4/2/2	0/0/2	2
H_6_16	14.9	0	2593	4/2/3	0/2/2	1
H_8_8	21.3	0	7817	3/2/2	0/2/1	0
H_8_10	18.5	0	16044	3/2/2	1/2/1	0
H_8_12	21.9	4.7	36000	3/2/3	1/0/2	2
H_8_14	17.4	0	33592	4/2/3	0/1/2	3
H_8_16	22.5	12.6	36000	4/3/3	1/0/3	2
H_10_10	18.8	0	21416	3/2/2	0/1/2	3
H_10_12	26.9	15.5	36000	4/3/3	1/0/2	1
H_10_14	25.7	17.1	36000	7/4/3	0/2/2	5
H_10_16	25.2	16.6	36000	4/2/3	0/0/3	4

is clearly seen on the larger instances.

All solutions for the feeder networks contain at least one butterfly route, where the hub is visited twice on a route. For the hub-and-spoke networks, about one third of all solutions contain routes that visit at least one port twice; all of these are found in solutions that are proven optimal.

Only considering the instances that are solved to proven optimality in Table 6, the model has been run with a single layer to make a comparison between simple and more complex service structures similar to the analysis in Thun et al. (2017). The results are presented in Table 7. The table shows the name of the instance, the relative increase in optimal objective value calculated as $QL = 100 \cdot \overline{z}^1_{IP}/\overline{z}^2_{IP}$ where $\overline{z}^l_{IP}$ is the optimal objective value of the model with l layers and the relative time used calculated as $TR = 100 \cdot T^1/T^2$ where T^l is the solution time of the model with l layers. In 10 out of 16 instances, the optimal objective value is better when using two layers. The difference in optimal objective value ranges between 0 and 31%, with an average increase of 4%. Weighting the difference with the absolute objective values gives an average increase of 2%, indicating that the saving from using two layers decreases on the larger instances tested. The advantage with the single layer model is the solution time. On the instances that are solved to proven optimality, the solution time is less than 3% of the time for the two layers, and three more instances are solved to proven optimality.

Table 7
Comparing the single layer and two layer models

Inst.	QL	TR	Inst.	QL	TR
F_4_6	113.0	22.7	H_6_10	104.5	42.9
F_6_10	131.2	77.4	H_6_12	100.8	1.5
F_8_14	112.9	9.7	H_6_14	100.6	1.6
H_4_4	100.9	83.3	H_6_16	100.0	4.9
H_4_6	101.8	43.9	H_8_8	100.0	5.1
H_4_8	100.0	41.8	H_8_10	100.3	0.9
H_6_6	100.0	22.6	H_8_14	101.9	1.3
H_6_8	102.1	19.1	H_10_10	100.0	2.8

The instance H_8_14 can serve as a case for a detailed analysis. Three routes are used in the optimal solution to the two layer model, one route is assigned middle sized vessels and two are assigned large vessels. The route for the middle sized vessels is $2 \rightarrow 3 \rightarrow 4 \rightarrow 7 \rightarrow 6 \rightarrow 7 \rightarrow 3 \rightarrow 2$. We see that ports 3 and 7 are visited twice and the route changes layers at port 6 and 2. Both routes for the large vessels are simple cycles. There are in total 304 containers transshipped at three different ports. The model has also been run with a single layer to make a comparison between simple and more complex service structures. The optimal solution in the one layer case needs four routes; a route assigned to the smallest vessels is also needed. The cost increases by about 2% and there are 878 containers transshipped at four different ports. Even though the difference in cost is not big, the savings by using more complex service structures can turn a red bottom line black in an industry with small margins.

As a final comparison we converted the Baltic instance in the LINER-LIB-2012 benchmark suite to an instance for this problem. The instance was run using the design speed of the vessels. The Baltic instance is similar in size to the F_12_22, and the root node gap and the final gap are 22.6% and 17.9% respectively. This corresponds well with the gaps for the F_12_22 instance and indicates that the instances generated here are roughly as complex as the instances in the LINER-LIB-2012 benchmark suite.

4. Conclusions

We have proposed a new formulation of the liner shipping network design problem based on a two layer network structure. The formulation takes into account the cost of transshipment and allows for complex service structures. A new set of small instances with up to twelve ports based on information from the LINER-LIB-2012 benchmark suite has been developed and the formulation has been tested on these instances. The computational study confirms that the liner shipping network design problem is a very complex problem. The proposed improvements of the formulation are shown to have a positive effect. Instances with up to 10 ports are solved to proven optimality.

Acknowledgments

This work was partly supported by the Research Council of Norway through the DOMinant II project. This support is gratefully acknowledged. We also thank the reviewers for there help in improving the paper.

References

R. Agarwal and Ö. Ergun. ”Ship scheduling and network design for cargo routing in liner shipping”. *Transportation Science*, 42:175–196, 2008.

J. Alvarez. ”Joint routing and deployment of a fleet of container vessels”. *Maritime Economics & Logistics*, pages 186–208, 2009.

B. D. Brouer, J. F. Alvarez, C. E. M. 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, 2014a.

B. D. Brouer, G. Desaulniers, and D. Pisinger. A matheuristic for the liner shipping network design problem. *Transportation Research Part E: Logistics and Transportation Review*, 72:42–59, 2014b.

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

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

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, 2012.

M. Desrochers and G. Laporte. Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, 10(1):27–36, 1991.

Drewry. Seaborne trade annual report 2013. Technical report, Drewry Maritime Research, 2014.

IHS Global Insight. ”Valuation of the liner shipping industry - Economic contribution and Liner Industry Operations”, 2009. URL `http://www.worldshipping.org/pdf/Liner_Industry_Valuation_Study.pdf`.

M. K. Jepsen, B. D. Brouer, C. E. M. Plum, D. Pisinger, and M. M. Sigurd. A path based model for a green liner shipping network design problem. In *Proceedings of The International MultiConference of Engineers and Computer Scientists*, volume 2, pages 1379–1384, 2011.

G. Laporte, M. Gendreau, J.-Y. Potvin, and F. Semet. Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research*, 7(4):285–300, 2000.

S.A. Lawrence. *”International sea transport: the years ahead”*. Lexington Books Lexington, Mass, 1972.

F. Margot. Symmetry in integer linear programming. In M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, editors, *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, pages 647–686. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

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.

C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4):326–329, 1960.

J. Mulder and R. Dekker. Methods for strategic liner shipping network design. *European Journal of Operational Research*, 235(2):367–377, 2014.

C. E. M. Plum, D. Pisinger, and M. M. Sigurd. A service flow model for the liner shipping network design problem. *European Journal of Operational Research*, 235(2):378–386, 2014.

L. B. Reinhardt and D. Pisinger. ”A branch and cut algorithm for the container shipping network design problem”. *Flexible Services Manufacturing Journal*, pages 349–374, 2011.

A. Santini, C. E. M. Plum, and Stefan Ropke. A branch-and-price approach to the feeder network design problem. *European Journal of Operational Research*, 264(2):607–622, 2018.

K. Thun, H. Andersson, and M. Christiansen. Analyzing complex service structures in liner shipping network design. *Flexible Services and Manufacturing Journal*, 29(3):535–552, 2017.

N. K. Tran and H. D. Haasis. Literature survey of network optimization in container liner shipping. *Flexible Services and Manufacturing Journal*, 27(2-3):139–179, 2015.

L. A. Wolsey. *Integer programming*. Wiley-Interscience, New York, NY, USA, 1998.