



Liner shipping network design

Christiansen, Marielle; Hellsten, Erik Orm; Pisinger, David; Sacramento, David; Vilhelmsen, Charlotte

Published in:
European Journal of Operational Research

Link to article, DOI:
[10.1016/j.ejor.2019.09.057](https://doi.org/10.1016/j.ejor.2019.09.057)

Publication date:
2020

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Christiansen, M., Hellsten, E. O., Pisinger, D., Sacramento, D., & Vilhelmsen, C. (2020). Liner shipping network design. *European Journal of Operational Research*, 286. <https://doi.org/10.1016/j.ejor.2019.09.057>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Liner Shipping Network Design

Marielle Christiansen¹, Erik Hellsten², David Pisinger^{2,3}, David Sacramento², Charlotte Vilhelmsen²

1. Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Trondheim, Norway

2. DTU Management, Technical University of Denmark, Akademivej 358, DK-2800 Kgs. Lyngby, Denmark

3. Corresponding author

Abstract The maritime industry is one of the greenest modes of transportation, taking care of almost 90 percent of the global trade. The maritime container business revolves around liner shipping, which consists of container vessels sailing on fixed itineraries. For the last 20 years, there has been an increasing number of publications regarding how to design such fixed routes (services), to ensure a high level of service while minimizing operational costs and environmental impact. The *liner shipping network design problem* can briefly be described as follows: Given a set of demands (defined by origin, destination, time limit) and a set of vessels with variable capacity, the task is to design a set of weekly services, assign vessels to the services, and flow the demand through the resulting network such that it arrives within the stated time constraints. The objective is to maximize revenue of transported demand subtracting the operational costs. We present an in-depth literature overview of existing models and solution methods for liner shipping network design, and discuss the four main families of solution methods: integrated mixed integer programming models; two-stage algorithms designing services in the first step and flowing containers in the second step; two-stage algorithms first flowing containers and then designing services; and finally algorithms for selecting a subset of proposed candidate services. We end the presentation by comparing the performance of leading algorithms using the public LINER-LIB instances. The paper is concluded by discussing future trends in liner shipping, indicating directions for future research.

Keywords OR in Maritime Industry; Network Design; Mixed-Integer-Programming; Heuristics; Multi-commodity Flow Problem

1 Introduction

Maritime transportation is fundamental to the world trade and globalisation, since it enables carrying large volumes at relatively low costs. It is estimated that around 90 percent of the world trade is carried by the international shipping industry. Maritime transportation is, furthermore, generally considered safe. Losses caused by incidents during transport by sea have dropped steadily, and are at the lowest values in decades. Additionally, efficient port structures make it possible to combine sea transportation with other land-based modes of transportation.

Roughly speaking, liner shipping can be split into ships designed for Lift-on/Lift-off (LoLo) operations, and Roll-on/Roll-off (RoRo) operations. In LoLo operations, quay cranes located on the docks are used to load and unload the ship's containers, while RoRo operations are designed to carry cargo that can be rolled on and off the ship.

In this paper, we mainly focus on *containerised* liner shipping network design, i.e. networks using LoLo operations, though we also briefly describe RoRo liner shipping. Although we have tried to cover most of the relevant papers dealing with containerised liner shipping network design, we have chosen to focus on models and algorithms that seem to be applicable in practice. This means that the selected papers

either have reported results on designing real-life networks, or later papers have used the framework to reach the same goal.

This paper is organized as follows: The rest of this section gives a brief introduction to containerised liner shipping, RoRo liner shipping, and network design. We also introduce the LINER-LIB test instances for network design in containerised liner shipping. The following sections are focused on containerised liner shipping. In Section 2 we discuss the challenges in designing a liner shipping network, and show that algorithms can roughly be split into four different families, which will be presented in the following sections. In Section 3 we give an overview of integrated Mixed Integer Programming (MIP) models, while Section 4 studies two-stage algorithms where the services are constructed in a first step, and containers are flowed through the resulting network in the second step. Section 5 considers algorithms for selecting a subset of proposed candidate services. In Section 6, we consider algorithms based on first flowing containers, and then designing services. Section 7 reports computational results for the LINER-LIB instances. The paper is concluded in Section 8 with a short discussion of future trends and challenges. Finally, an overview of the notation used throughout the paper is found in Appendix A. Parts of this paper are based on the survey by Hellsten et al. (2018) and the book chapter Christiansen et al. (2019).

1.1 Containerised Liner Shipping

The liner shipping industry is a vital part of the global economy, constituting one of the cheapest modes of cargo transport. During the last three decades, the volume of containerised cargo has grown by more than 8% per year, and more than 5,150 container vessels were in operation worldwide in 2017. Standard containers come in two different sizes, twenty and forty feet, which have given rise to the standard measures of containerised cargo, *twenty foot equivalent units* (TEU) and *forty foot equivalent units* (FFE). The largest vessels carry more than 20,000 TEU and during 2016, a container volume of around 140,000,000 TEU was estimated to pass through the vast liner shipping network (Unctad, 2017a,b). Clearly, any improvement in the network design in the liner shipping industry will correspond to enormous savings. This section will briefly introduce the liner shipping business and then the corresponding network design problem, along with its variants and various solution and modelling approaches presented in the literature.

The liner shipping industry is built up by so called *services*. A service is a fixed cyclic itinerary, sailed by a number of similar vessels. Services usually have weekly or biweekly departures to add consistency and regularity for the customers. The vessels are operated by shipping companies called carriers, where the largest carriers operate over 600 vessels. As larger vessels are more energy efficient, see Figure 1, the trend is to build ever larger vessels. To efficiently utilise those very large liner vessels, each region typically has a few larger ports, called *hubs*, where the liner vessels pick up and deliver containers. From the hubs, the containers are then transported to other ports by smaller, more flexible, so called *feeder vessels*. The act of transferring containers from one vessel to another in a port is called *transshipping*. Transshipments occur both between larger vessels and smaller vessels, but also between larger vessels when no suitable service connects the origin and destination hub. While transshipments add flexibility, they tend to be costly, as the cargo needs to be unloaded, stored until the arrival of the new vessel and then reloaded again. Finally, *cabotage rules* need to be taken into account. To protect the national trade business, many countries forbid foreign carriers to ship cargo between two ports within the country as well as other restrictions. See Brouer et al. (2014a) and Zheng et al. (2014) for examples of such rules.

The major costs for the carriers are vessel acquirement and bunker fuel. However, other costs, like canal fees, port costs and transshipment costs, are also highly significant. The fuel consumption is frequently estimated as a cubic function of the speed, as seen in Figure 2. As the speed has such an impact on the fuel consumption, *slow steaming* is often used to reduce the consumption. Especially after the financial crisis in 2008, maritime shipping companies implemented slow steaming policies for cost-cutting

Cost per 1000 container miles

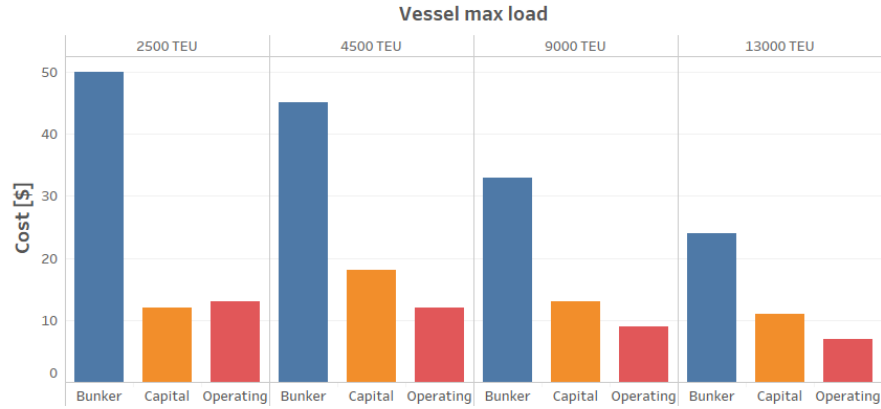


Fig. 1. Estimated cost per 1,000 container miles for different vessel sizes. The vessels are assumed to sail at 19 knots and the bunker fuel price is estimated as 750 \$/tonne. We see that bunker represents the largest cost and that transporting containers on larger vessels requires significantly less fuel. Source: Germanische Lloyd (2017).

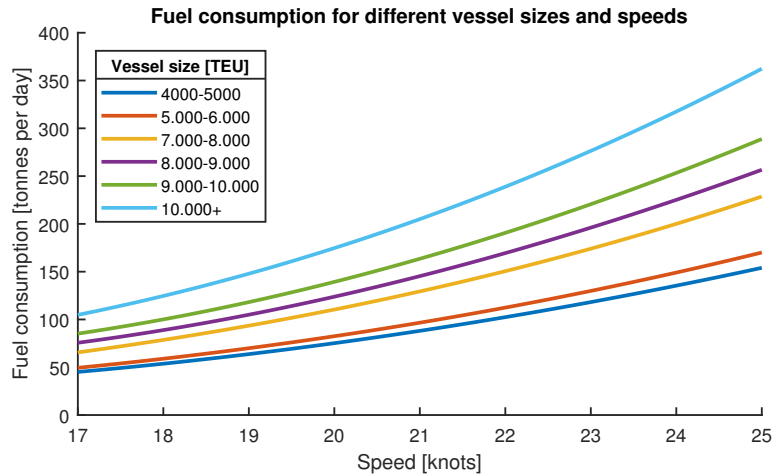


Fig. 2. Estimated fuel consumption as a function of steaming speed and vessel size. Source: Notteboom and Vernimmen (2009).

purposes. The obvious drawback of slow steaming is that more vessels are required to transport the same amount of cargo and also, that transit times become longer, yielding a lower level of service for the customers. In general, services have two directions, *head-* and *back-haul*, where most of the cargo is transported in the head haul direction. A good example of this is the trade between Asia and Europe, where most of the goods are delivered from Asia to Europe. In this case, vessels are slow steaming in the back-haul direction where less customers are affected by the transit time.

1.2 Containerised Liner Shipping Network Design Problem

The Liner Shipping Network Design Problem (LSNDP) can be defined as follows: Given a collection of ports, a fleet of container vessels and a group of origin-destination demands, a set of services is

constructed for the container vessels such that the overall operational expenses are minimised, while ensuring that all demands can be routed through the resulting network from their origin to their destination, respecting the capacity of the vessels.

In the following we present some notation of the LSNDP that will be used throughout the paper, introducing the necessary notation when required. A complete table of notation can be found in Table 5 in Appendix A. For a complete model, where the LSNDP is described in detail as well as a presentation of the liner-shipping business along with a description of its main assets and infrastructure, see Brouer et al. (2014a).

The set of ports is denoted by N and represents the set of physical ports in the problem. The set of arcs A represents all possible sailings between ports. The set of demands or commodities is denoted by K and for each commodity $k \in K$, there is an origin port o_k , a destination port d_k , as well as a quantity q_k measured in twenty-foot equivalent units (TEU). Furthermore, the corresponding unit-cost for transporting a unit of commodity k through arc $(i, j) \in A$ is defined as c_{ij}^k . Finally, the set V denotes the set of *vessel classes*. For each vessel class $v \in V$ there is a corresponding cargo capacity, u_v , measured in number of TEUs, an available fleet quantity, m_v , as well as additional speed limitations and fuel consumption parameters. Furthermore, for convenience, the demand of the commodities in each port $i \in N$ is defined as:

$$\xi_i^k = \begin{cases} q_k & \text{if port } i \text{ is the origin port of commodity } k \\ -q_k & \text{if port } i \text{ is the destination port of commodity } k \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

There is a limited fleet of container vessels, but not all vessels need to be used. The deployment of a vessel $v \in V$ has an associated *charter cost* c^v . Additionally, there are other costs related to the resulting network, such as the *sailing cost* c_{ij}^v associated with each vessel and each arc, which is given as a combination of the port call cost and the fuel consumption for the corresponding leg. When containers of commodity $k \in K$ are transferred from one vessel to another in port $i \in N$, there is a *transshipment cost* c_{ik}^T for each container. Furthermore, there is an associated sailing time t_{ij}^v for each container vessel v sailing between ports i and j , which is calculated from the vessel's design speed and the distance between the ports. Moreover, each port $i \in N$ has an associated berthing time b_i .

One of the main traits of the liner shipping industry is the regular operation of services under a pre-established schedule. Sometimes it is possible to define the set of candidate services in advance. In these cases, let S be the set of feasible services in the model. Notice that S can be exponentially large. Each service $s \in S$ has an associated operational cost c_s . As the set of services is defined beforehand, the operational cost is given as a combination of the sailing cost of the arcs on the service route and the corresponding port-call costs. Moreover, it is demanded that all services should have *weekly operations*, meaning that if a round trip takes eight weeks to complete, then eight similar vessels need to be deployed to the service in order to ensure that each port is visited once a week. Therefore, the required number of vessels from vessel class v , to maintain the weekly frequency, is defined as m_v^s . In addition, services must be cyclic, visiting a sequence of ports while respecting the weekly departures, but the structure of the services can be divided into several types according to the number of times a port is visited during the service. A *simple service* or a *circular service* visits each port in the service exactly once. However, a service is often allowed to be non-simple, meaning that a port can be visited several times, as this may improve transit times. Nodes (or ports in the sequence) that are visited several times in the service are denoted *butterfly nodes*. We can distinguish between different types of services; a service containing a single butterfly node defines a *butterfly service*, whereas a service visiting all ports twice in both directions, but in reverse order, is defined as a *pendulum service*. Furthermore, a service in which any node can be visited multiple times is defined as a *complex service*. Examples of the different type of services for some European ports are illustrated in Figures 3, 4, 5 and 6.

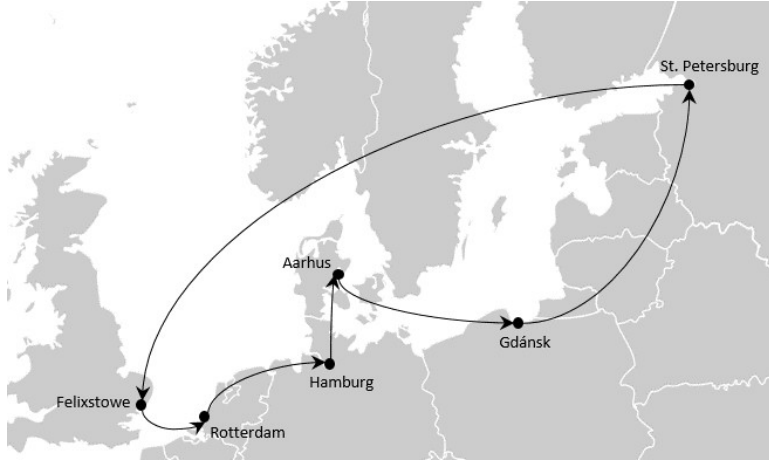


Fig. 3. Example of a simple service, where each port is visited exactly once.

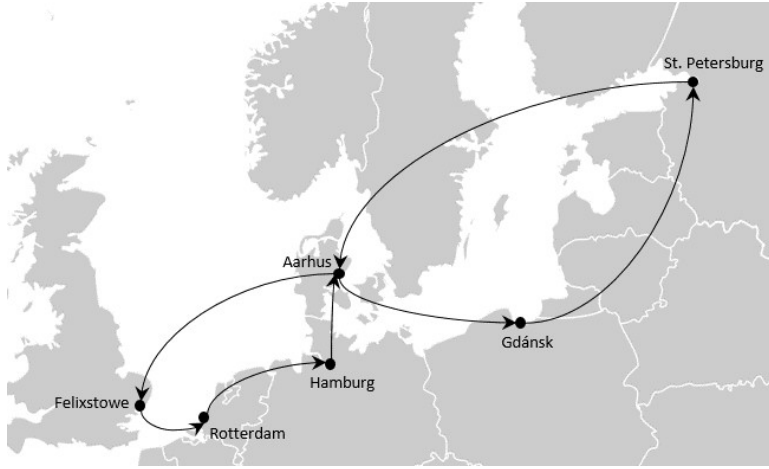


Fig. 4. Example of a butterfly service, where Aarhus is the butterfly node.

The variants of the LSNDP, which have been studied in the literature, vary mainly in the following four respects:

- *Transit time constraints.* As described above, the transit time of each demand has an associated time limit that must be respected. If the transit time is not respected, perishable goods may become spoiled.
- *Transshipment costs.* The costs of transshipments are a significant part of the operational costs (Karsten (2015)), so it is generally important to represent these costs properly in the model.
- *Rejected demands.* Although the standard formulation of LSNDP states that all demands must be flowed through the network, many models allow rejection of demands by imposing a penalty.
- *Speed optimisation.* There are three main approaches to model regarding speed optimisation: Models which have constant speed for all services, models which choose a speed for each service, and models which choose a speed on each individual leg in each service. As the fuel consumption depends non-linearly on the speed, it is common to choose between a number of discrete speed alternatives, each with a corresponding cost.

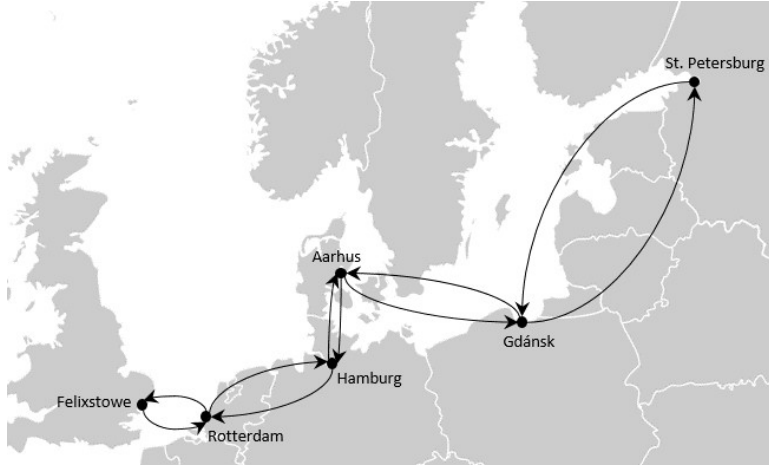


Fig. 5. Example of a pendulum service, where each port can be visited both on the head-haul and back-haul trip.

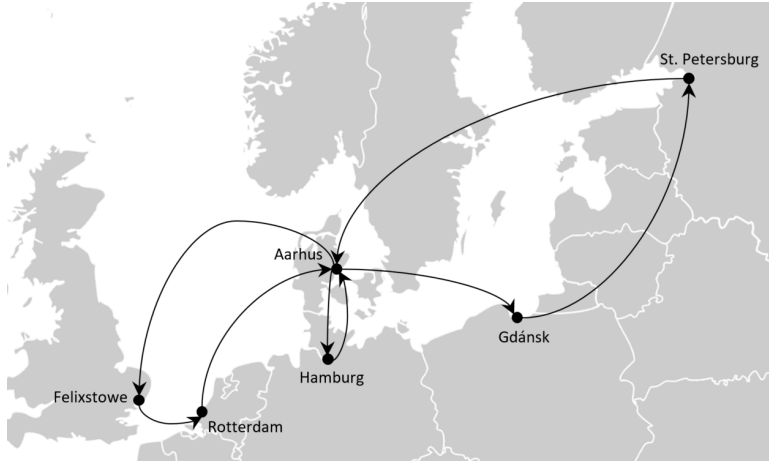


Fig. 6. Example of a complex service, where Aarhus is a butterfly node that is visited three times.

Most models for LSNDP design the network without a specific schedule. Hence the service for each vessel is defined, but not the exact day of arrival/departure. This is typically done in a later step, where port availabilities are negotiated and transshipment times at ports are adjusted.

For a detailed review of the research on liner shipping optimisation problems, see the survey papers Ronen (1983, 1993), Christiansen et al. (2004, 2013), Kjeldsen (2011), Meng et al. (2014), Tran and Haasis (2015), Brouer et al. (2016, 2017), and Lee and Song (2017).

1.3 RoRo Network Design Problem

Roll-on Roll-off (RoRo) shipping is an important segment within liner shipping. The RoRo ships are vessels designed to carry wheeled cargo, such as cars, trucks, semi-trailer trucks, and railroad cars, that can be driven on and off the ship on their own wheels. In addition, RoRo ships may carry complex cargo that is placed on trolleys and rolled on and off the ships, such as boats, helicopters, and heavy plant equipment. RoRo shipping is often the only viable method of ocean freight transportation for these oversized vehicles, as they may not fit in standard containers. There exist various types of RoRo ships,

such as ferries, cruise ferries, cargo ships, and barges. In this subsection, we consider the RoRo ships used for transporting cars, trucks and complex general cargo across oceans known as Pure Car Carriers (PCC), Pure Truck & Car Carriers (PCTC) and general RoRo ships, respectively. A typical PCTC has a carrying capacity in the range of **5,500 to 8,000** RT43. Here, RT43 is a capacity measure in the RoRo business and corresponds to the size of a 1966 Toyota Corona. The world fleet of RoRo ships consists of around **5,000** ships with a total capacity of more than 24 million deadweight tons (ISL, 2016).

The trades to be serviced in RoRo shipping are usually designed based on a large number of contracts for transportation of cargo between the different port pairs along a trade. Hence, trade routes are defined as transportation arrangements from one geographical region to another, where the world is divided into a number of geographical regions. Each trade route has a number of loading ports in one region and a number of discharging ports in the other. In Figure 7, two trade routes are illustrated by solid lines and the ports are shown as filled circles. After a ship has sailed one voyage on a trade route it often needs to reposition to start on the next one due to trade imbalances. This repositioning means ballast sailing, i.e. sailing without cargo, which of course should be reduced as much as possible. The ballast sailing between the two trade routes in Figure 7 is illustrated by a dashed line. Differences in contractual requirements and a variety in the types of cargo transported on the various trade routes may restrict which vessels that can be assigned to a particular trade route, regarding both capacity and vessel type.

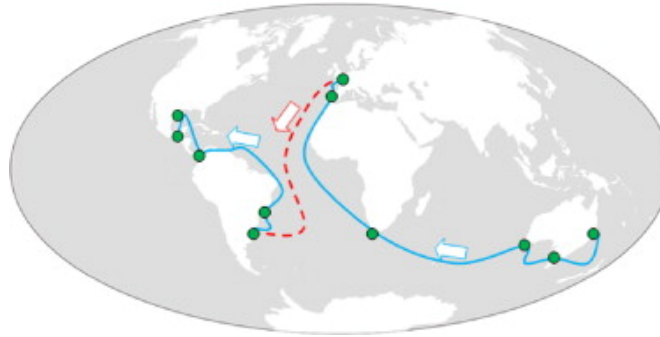


Fig. 7. Illustration of two trade routes sailed in sequence, Oceania to Europe via South Africa, and South America to North America, with associated ballast sailing from Europe to South America. Source: (Andersson et al., 2015).

Each trade route is sailed regularly, for example weekly, fortnightly, 3 times per months, depending on demand and contractual obligations. Each sailing on a trade route is called a voyage, and normally there is a time window for start sailing a voyage. Due to contractual obligations, these voyages are mandatory and must be covered either by a ship in the RoRo shipping company's own fleet or by a chartered ship. A RoRo shipping company owns and operates a heterogeneous fleet of ships having different cargo capacities, sailing speed ranges, and bunker consumption profiles, and serve a given set of trade routes.

Planning problems within RoRo shipping are far less studied in the OR literature compared to container shipping. However, this segment has received increased attention the last decade. Pantuso et al. (2015) consider strategic planning issues involving decisions regarding fleet size and mix. On the tactical planning level, the fleet deployment problem consists of assigning ships in the fleet to voyages that must be performed repeatedly on given trade routes. In addition to the ship-voyage assignment, the results from fleet deployment are sailing routes for the ships in the fleet, i.e. each ship is assigned a sequence of voyages to perform, possibly with ballast (empty) sailing between the last port call of one voyage and the first on the next. Fagerholt et al. (2009) present a mixed integer programming model for the fleet

deployment problem in RoRo shipping, and Andersson et al. (2015) extend this model for a real fleet deployment problem by including speed as a decision variable.

Large RoRo shipping companies are well placed to offer end-to-end integrated logistics services to car manufacturers engaged in international trade of vehicles. The fleet deployment planning in RoRo shipping can achieve better results by integrating inventory management of cargos at ports where the cargo is produced and/or demanded. A few studies including fleet deployment and inventory management in RoRo shipping exist to show the potential of combining these planning tasks even though most RoRo shipping companies perform separate planning today; Chandra et al. (2015, 2016), Dong et al. (2017).

The operations within RoRo shipping deviates from container shipping in several ways as well as the cargo and ships. In container shipping each ship is normally assigned to a single route, while in RoRo shipping a ship may sail several trade routes during a planning horizon. Ship types instead of individual ships are often considered in container shipping, while in RoRo shipping a route for each ship is determined. This also means that in RoRo shipping each trade route may be serviced by different ship types. Furthermore, there is a great variation in when to start each voyage, as well as when and how often to visit each port along the trade. Therefore, existing studies within fleet deployment in RoRo shipping have used time windows for when each voyage along each trade should start. This flexibility is in contrast to container shipping, as each service is usually served on a strict weekly basis, and each voyage along the trade visits all ports in the same order. Finally, transshipment rarely exist in RoRo shipping in contrast to container shipping, and none of the relevant RoRo studies includes this aspect.

1.4 The LINER-LIB test instances

In order to make it easier to compare algorithms for liner shipping network design, Brouer et al. (2014a) introduced the LINER-LIB benchmark suite. The test instances in LINER-LIB are based on real-life data from leading shipping companies along with several other industry and public stakeholders. The benchmark suite contains data on ports including port call cost, cargo handling cost and draft restrictions, distances between ports considering draft and canal traversal, vessel related data for capacity, cost, speed interval and bunker consumption, and finally a commodity set with quantities, revenue, and maximal transit time. The commodity data is intended to reflect the differentiated revenue associated with the current imbalance of world trade.

The LINER-LIB benchmark suite consists of seven instances described in Brouer et al. (2014a) and is available at <http://www.linerlib.org>. The instances range from smaller networks suitable for being solved by exact solution methods to large scale instances spanning the globe. Table 1 gives an overview of these instances.

Instance	Category	$ N $	$ K $	$ V $	min v	max v
Baltic	Single-hub	12	22	2	5	7
WestAfrica	Single-hub	19	38	2	33	51
Mediterranean	Multi-hub	39	369	3	15	25
Pacific	Trade-Lane	45	722	4	81	119
AsiaEurope	Trade-Lane	111	4,000	6	140	212
WorldSmall	Multi-hub	47	1,764	6	209	317
WorldLarge	Multi-hub	197	9,630	6	401	601

Table 1. The seven test instances included in LINER-LIB with indication of the number of ports ($|N|$), the number of origin-destination pairs ($|K|$), the number of vessel classes ($|V|$), the minimum (min v) and maximum number of vessels (max v) in each class.

Each of the instances can be used in a low, base, and high capacity case depending on the fleet of the instance. For the low capacity case the fleet quantity and the weekly vessel costs are adjusted to fewer vessels with a higher vessel cost, while for the high capacity case the adjustments are reversed.

Currently, most papers only report results for the base capacity case. Furthermore, most often only the six first instances are considered, with Krogsgaard et al. (2018) being the only to report results for the *WorldLarge* instance.

2 Overview of models and algorithms

Designing a liner shipping network is a difficult task, embracing several decisions: Not only do we need to construct the individual services, but we should also deploy vessels of the right size to each service and ensure that there is sufficient capacity in the network to transport all containers from their origin to their destinations. Designing the individual services is an *NP*-hard problem, as proved in Brouer et al. (2014a). Furthermore, routing the containers through a given network subject to time constraints for each container, can be recognised as a time-constrained multi-commodity flow problem, which is also *NP*-hard (Karsten et al., 2015).

The problem is further complicated by the fact that ports are often visited several times in the same service. This is obviously the case for pendulum services where a vessel is sailing back and forth along the same service, but multiple visits to a port (typically a hub) often takes place to ensure that containers quickly can be transshipped to other services. However, formulating the problem with multiple visits to a port as a MIP model becomes more difficult.

Finally, one should notice that transshipment costs represent the majority of the cost of routing the containers through the network according to Psaraftis and Kontovas (2015). It is therefore important to carefully model which containers might be transshipped and at which costs. This adds further complexity to the problem, and makes a graph or MIP formulation huge and difficult to solve.

Algorithms for liner shipping network design can roughly be divided into the following four groups:

- *MIP-based algorithms.* These algorithms are based on a unified MIP model that designs services and flows containers through the resulting network. In order to handle this task, two sets of variables are needed: Variables to select arcs in a service, and variables to denote the flow on each arc. If multiple visits to a node are allowed (butterfly nodes) then an additional index is needed to indicate the visit number at each node. *Several MIP-based models have been presented in the literature, including Álvarez (2009), Reinhardt and Pisinger (2012), Plum et al. (2014a,b), and Wang and Meng (2014).*
- *Two-stage algorithms.* As the name suggests, these algorithms solve the problem in two steps: Designing the services, and flowing containers through the resulting network. Frequently, these algorithms contain a feed-back mechanism, where output from the second-stage flow model is used as input to improve the services in the first stage. *Successful applications of this approach include Agarwal and Ergun (2008), Álvarez (2009), Brouer et al. (2014a,b), Mulder and Dekker (2014), Karsten et al. (2017b), Thun et al. (2017), and Neamatian Monemi and Gelareh (2017).*
- *Subset of routes.* Both Meng and Wang (2011b) and Balakrishnan and Karsten (2017) suggest a heuristic method for generating a network by having a list of candidate services as input. The idea behind these algorithms is to use the experience from existing planners to design a large number of promising candidate services. The algorithm then selects a subset of the candidate services to form a network. Many shipping companies and customers do not want the network to be completely restructured, in which case proposing small variations to each service may be a sensible method.
- *Backbone flow.* It can be difficult to design the individual services without knowing how the containers will flow through the network. Hence, another approach is to reverse the order of the subproblems in the two-stage algorithms, and start by finding an initial flow (a so-called backbone network) where

cargo is flowed through a complete network with all connections between ports available. The connections are priced such that they are expensive at low loads and cheap at high loads, in order to make the cargo gather at few connections. The initial flow can be seen as an accomplishment of the *physical internet* (Montreuil, 2011) where point-to-point transport has been replaced by multi-segment intermodal transport. A successful application of the backbone network idea was presented in Krogsgaard et al. (2018).

Many of the MIP-based algorithms can in principle solve the LSNDP to optimality. However, due to the intrinsic complexity, only small instances can be solved to proven optimality within a reasonable time frame. The subset-of-services based algorithms also solve the problem to optimality given that only the proposed candidate services are valid. In practice, however, there may be an exponential number of valid services, and we cannot expect to get all services as input. The two-stage algorithms and backbone-network algorithms are both heuristics, since they first solve one stage, and then optimise the second stage with the first-stage decisions fixed.

Table 2 gives an overview of the most important papers that have presented algorithms for the LSNDP. The papers are ordered according to year of publication. For each paper we indicate which variant of LSNDP is considered (transit time constraints, transshipment costs, rejected demand, and speed optimisation). Moreover it is indicated which solution method is used, and whether services are simple, butterfly, or general.

It is generally seen that there is a large variation in which variant of the problem authors consider, and a large spectrum of solution methods are being developed. This somehow indicates that the LSNDP is still a quite young research area under development. Moreover, most work on the LSNDP is tightly linked to industrial applications, meaning that constraints and assumptions need to be adjusted according to the setup of the the industrial collaboration partners.

Paper	Transit time constraints	Transshipment costs	Reject demand	Speed optimisation	Type of service	Solution method
Agarwal and Ergun (2008)	No	Yes	Yes	Legs	General	Two-phase, Column Generation and Benders
Álvarez (2009)	No	Yes	Yes	Services	Butterfly	Two-phase, Column generation heuristic
Gelareh et al. (2010)	Yes	Yes	Yes ¹	Design	Circular	MILP, Lagrangian decomposition method
Meng and Wang (2011a)	Yes ²	Yes ³	No	Legs	General	MILP (Non-Linear), Branch-and-bound based ϵ -optimal algorithm
Meng and Wang (2011b)	No ⁴	Yes	No	Design	General	MILP, exact solution
Reinhardt and Pisinger (2012)	No	Yes	No	Design	Butterfly	MILP, exact solution, branch-and-cut
Plum et al. (2014a)	Yes	No	Yes	Design	Circular	MILP, branch-and-cut-and-price
Plum et al. (2014b)	No	Yes	Yes	Design	General	MILP, exact solution
Brouer et al. (2014a)	No	Yes	Yes	Services	Butterfly	Two-phase, heuristic column generation, tabu search
Brouer et al. (2014b)	No	Yes	Yes	Services	Butterfly	Two-phase, Matheuristic
Wang and Meng (2014)	Yes	No	Yes	Design	General	MILP, Column Generation heuristic method
Mulder and Dekker (2014)	No	Yes	Yes	Services	General	Two-phase, Genetic Algorithm, Dissagregation techniques
Brouer et al. (2015)	Yes	Yes	Yes	Services	Butterfly	Two-phase, Matheuristic
Neamatian Monemi and Gelareh (2017)	No	Yes	Yes	Legs	General	Two-phase, Benders
Balakrishnan and Karsten (2017)	No ⁴	Yes	Yes	Design	General	Subset of services, heuristic solution
Karsten et al. (2017a)	Yes	Yes	Yes	Legs	Butterfly	Two-phase, Matheuristic
Karsten et al. (2017b)	Yes	Yes	Yes	Services	Butterfly	Two-phase, Matheuristic
Thun et al. (2017)	No	Yes	No	Design	General	MILP, Branch-and-Price
Krogsgaard et al. (2018)	No	Yes	Yes	Services	General	Backbone Flow, Variable Neighborhood Search, Lagrange heuristic
Koza et al. (2019)	Yes	Yes	Yes	Legs	General	Integrated LSNDP and scheduling, column generation, matheuristic

Table 2. Overview of selected papers, presenting algorithms for LSNDP, sorted according to year of publication. **Transit time constraints** indicate whether each commodity has an upper limit on the transit time. **Transshipment costs** states whether these costs are taken into account. **Reject demand** indicates whether demands may be rejected by paying a penalty. **Speed optimisation** is either: Design (all vessels are sailing at design speed), services (variable speed on services but all legs in a service have the same speed), legs (different speed can be used on any leg). **Type of services** can be: Circular (every port is visited once in a rotation), butterfly (only one butterfly node in each service), general (several butterfly nodes are allowed). **Solution Method** first states which of the four main categories the algorithm belongs to (MILP model, two-phase, subset of services, backbone flow), and then briefly elaborates the methods used.

Additional comments:

¹ Attraction function defined in the objective function aiming to maximise the customer attractiveness of the services.

² Maximum tolerable average transit time between port-calls.

³ Container handling cost at the port-calls.

⁴ Limited number of transshipments.

3 Mixed Integer Programming Models

Numerous decisions must be taken when designing a service network, among which the routing of containers, the fleet deployment and the service design stand out. Defining a problem that correctly accounts for all these details in real life is beyond the limited capacity of human planners. Therefore, the use of complex decision support tools to design liner networks can be of great use. In this section we present MIP and graph-based models to define the LSNDP. Different formulations are briefly introduced to model the network design problem in liner shipping and a summary of the main mathematical formulations proposed in the literature are presented under different assumptions.

3.1 Service formulation for LSNDP

In this section, a service flow formulation for the LSNDP is introduced, where the set of all feasible services is defined in advance for the model. This reduces the network design problem to the selection of feasible services.

Let us begin with introducing a basic service formulation. We use the terminology presented in Section 1.2 with the addition of the following definitions: Let $G = (N, A)$ be a directed graph. Define for each service s and for each arc (i, j) , the associated capacity u_{ij}^s . Finally, let x_{ij}^{ks} be a continuous variable denoting the amount of commodity k transported by service s on arc (i, j) , and y_s a binary variable for the selection of service s in the network. Now, the service formulation of the LSNDP can be expressed as:

$$\min \sum_{s \in S} c_s y_s + \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k \sum_{s \in S} x_{ij}^{ks} \quad (2a)$$

$$\text{s.t.} \quad \sum_{s \in S} \sum_{j: (i,j) \in A} x_{ij}^{ks} - \sum_{s \in S} \sum_{j: (j,i) \in A} x_{ji}^{ks} = \xi_i^k \quad i \in N, k \in K \quad (2b)$$

$$\sum_{k \in K} x_{ij}^{ks} \leq u_{ij}^s y_s \quad s \in S, (i, j) \in A \quad (2c)$$

$$\sum_{s \in S} m_v^s y_s \leq m_v \quad v \in V \quad (2d)$$

$$x_{ij}^{ks} \geq 0 \quad (i, j) \in A, k \in K, s \in S \quad (2e)$$

$$y_s \in \{0, 1\} \quad s \in S. \quad (2f)$$

The objective function (2a) minimises the total operational cost of the network. The first term accounts for the fixed cost of the selected services, whereas the second term constitutes the sailing cost of shipping the demand. Constraints (2b) are the flow conservation constraints, and the flow of commodities on the legs has to respect the capacity of the vessel deployed in the selected service s as formulated in constraints (2c). For each vessel class $v \in V$, constraints (2d) ensure that the number of deployed vessels on the selected services using vessel class v does not exceed the maximum availability m_v . Finally, the domain of the variables is defined in constraints (2e) and (2f).

The formulation allows to design networks considering only a subset of promising candidate services. Examples of papers using this formulation are Álvarez (2009), Meng and Wang (2011b) and Balakrishnan and Karsten (2017).

For a better utilisation of the capacity of the vessels, the model can handle the cargo rejection by defining extra continuous variables that account for the demand that is rejected by the liner company, and incurring a large penalty in the objective function. Moreover, if we only consider simple services, additional continuous variables f_i^{ks} could be added to account for transshipments, denoting the amount

of commodity k that is transshipped at port i from service s . The following constraints (3) could be added to model transshipments.

$$f_i^{ks} \geq \sum_{\substack{j \in N \\ j \neq i}} x_{ji}^{ks} - x_{ij}^{ks} \quad i \in P \setminus \{o_k, d_k\}, k \in K, s \in S. \quad (3)$$

Some of these approaches have been considered by Álvarez (2009), who studies an LSNDP at the tactical level, considering the joint routing and deployment of container vessels. The model includes many relevant parameters in the objective function to correctly represent the operational cost of the selected services over a tactical planning horizon, and it is one of the first formulation to consider transshipment when designing the shipping network.

The formulation presented in this paper is based on the set of all feasible services, which are given as a combination of a vessel class, an operating speed and a route structure. Therefore, it is possible to accommodate services that are proposed externally by the planners as well as services generated internally by a solution algorithm, meaning that any type of non-simple services, i.e. services that visit one or more ports several times during a service round, can be considered in the service set S . However, the model is unable to accurately calculate the transshipment cost of non-simple services. Finally, the model considers the fleet deployment of the available fleet, defining integer variables that control the amount of vessels deployed for a chosen service.

As the size of the problem increases, the number of feasible services in the problem grows exponentially, making the model intractable to solve. Álvarez (2009) proposes a tabu search algorithm combined with a column generation procedure to solve a case study with up to 120 ports.

3.2 Arc formulation for LSNDP

The main problem with a service-based formulation is that generating all services S is non-trivial, due to the high number of combinatorial possibilities. Therefore, an alternative compact formulation is introduced in this section, which is based on an arc formulation. We are no longer considering a set S of predefined services, but instead let S^v be an index set for the services of vessel class v , indexed by s .

We first present a basic mathematical model based on arc formulation. We again use the notation presented in Section 1.2, with small extensions. Let $G = (N, A)$ be a directed graph. Let x_{ij}^{ks} be a continuous variable denoting the flow of commodity k on arc (i, j) by service s , which is operated by vessel class v , and y_{ij}^s a binary variable for the selection of arc (i, j) in service s . The binary variable γ_i^s is equal to 1 if port i is the hub port in service s . Moreover, we define τ_i^s as a continuous variable representing the time of departure of service s , for vessel class v , from port i , and w_s as an integer variable indicating the number of vessels from class v needed to maintain the weekly frequency of service s . Then, the arc formulation of the LSNDP can be expressed as follows:

$$\min \sum_{v \in V} \sum_{s \in S^v} c^v w_s + \sum_{v \in V} \sum_{s \in S^v} \sum_{(i,j) \in A} c_{ij}^v y_{ij}^s + \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k \sum_{v \in V} \sum_{s \in S^v} x_{ij}^{ks} \quad (4a)$$

$$\text{s.t.} \quad \sum_{v \in V} \sum_{s \in S^v} \sum_{j: (i,j) \in A} x_{ij}^{ks} - \sum_{v \in V} \sum_{s \in S^v} \sum_{j: (j,i) \in A} x_{ji}^{ks} = \xi_i^k \quad i \in N, k \in K \quad (4b)$$

$$\sum_{j: (i,j) \in A} y_{ij}^s - \sum_{j: (j,i) \in A} y_{ji}^s = 0 \quad i \in N, v \in V, s \in S^v \quad (4c)$$

$$\sum_{i \in N} \gamma_i^s = 1 \quad v \in V, s \in S^v \quad (4d)$$

$$\sum_{k \in K} x_{ij}^{ks} \leq u_v y_{ij}^s \quad (i, j) \in A, v \in V, s \in S^v \quad (4e)$$

$$\tau_j^s \geq (\tau_i^s + t_{ij}^v + b_j)(1 - \gamma_j^s)y_{ij}^s \quad i, j \in N, v \in V, s \in S^v \quad (4f)$$

$$\sum_{(i,j) \in A} (t_{ij}^s + b_j)y_{ij}^s \leq 24 \cdot 7 \cdot w_s \quad v \in V, s \in S^v \quad (4g)$$

$$\sum_{s \in S^v} w_s \leq m_v \quad v \in V \quad (4h)$$

$$x_{ij}^{ks} \geq 0 \quad (i, j) \in A, k \in K, v \in V, s \in S^v \quad (4i)$$

$$y_{ij}^s \in \{0, 1\} \quad (i, j) \in A, v \in V, s \in S^v \quad (4j)$$

$$w_s \in \mathbb{Z}^+ \quad v \in V, s \in S^v \quad (4k)$$

$$\tau_i^s \geq 0 \quad i \in N, v \in V, s \in S^v \quad (4l)$$

The objective function (4a) minimises the cost of deploying the vessels and designing the services and the cost of transporting the containers through the network. The flow conservation constraints for the cargo variables are given in constraints (4b), whereas the flow conservation constraints for the routing variables are given in constraints (4c). Constraints (4d) ensure that there is only a single hub port for each service. The flow of cargo on an edge (i, j) cannot exceed the capacity u_v of a vessel class, as expressed in (4e). If the service does not use a given edge in the graph, i.e. $y_{ij}^s = 0$, then the capacity is zero. The time schedule constraints for the routing variables are given by the time variables in constraints (4f). Note that it is necessary to linearise these constraints, as they are non-linear. Moreover, these constraints also ensure the elimination of sub-tours when designing the liner network. The weekly frequency of the services and the deployment of the fleet are expressed by constraints (4g). The availability of the fleet is limited by constraints (4h). Finally, the domain of the variables is defined by constraints (4i)-(4l).

This model is a simple representation of the arc formulation for the LSNDP, and it is a fairly easy adaptation of a variant of the Vehicle Routing Problem (VRP) (Toth and Vigo, 2015). However, this model can be extended to incorporate the various constraints and considerations encountered in liner shipping.

An arc formulation with butterfly services

As argued by Agarwal and Ergun (2008), transshipment is the core of liner shipping. Hence, these operations should not be ignored when designing the shipping network. Reinhardt and Pisinger (2012) extend the literature by proposing a MIP model based on an arc-flow formulation, where the network design and fleet assignment are combined. The model can handle butterfly services as well as account for the transshipment cost in the butterfly nodes.

Next follows a description of this model, adapted to the notation presented in the Section 1.2, with small extensions. Let $G = (N, A)$ be a directed graph. Due to the heterogeneous fleet considered by the authors, the set V defines the set of vessels, instead of the set of vessel classes. Therefore, we consider each vessel v to belong to its own vessel class.

The design of the network is modelled with the binary variables y_{ij}^v for the utilisation of an arc (i, j) in the service for vessel v . Similarly, as proposed by Miller et al. (1960), positive integer variables e_{ij}^v are defined for enumerating the arcs used in the vessel service and avoid subtours in services. The binary variables γ_i^v and z_{ij}^v identify, respectively, the unique center-point, i.e. the butterfly node in the vessel service, and allow the possibility of identifying the first and last arc visiting the butterfly node. These variables are used for modelling the transshipment of cargo in butterfly nodes.

The routing of containers through the network is modelled with continuous variables x_{ij}^{kv} , and extra continuous variables are defined to count the amount of the transshipped containers in intermediate ports within the same service. Let the continuous variables f_j^{kv} define the amount of commodity k transshipped by vessel v at port j , while the continuous variables f_{jih}^{kv} denote the amount of commodity k , arriving at port i through arc (j, i) , in vessel v , not leaving in arc (i, h) .

Let t_{max} be the length of the time horizon. The deployment of a vessel is controlled by the binary variable λ^v , whereas the continuous variables τ_v limit the service length of the vessels. Then, the arc-flow model can be defined as:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k \sum_{v \in V} x_{ij}^{kv} + \sum_{k \in K} \sum_{i \in N} c_{ik}^T \sum_{v \in V} f_i^{kv} + \sum_{v \in V} c^v \lambda^v \quad (5a)$$

$$\text{s.t.} \quad \sum_{v \in V} \sum_{j: (i,j) \in A} x_{ij}^{kv} - \sum_{v \in V} \sum_{j: (j,i) \in A} x_{ji}^{kv} = \xi_i^k \quad i \in N, k \in K \quad (5b)$$

$$f_i^{kv} \geq \sum_{j: (i,j) \in A} x_{ij}^k - \sum_{j: (j,i) \in A} x_{ji}^k \quad k \in K, i \in N, v \in V \quad (5c)$$

$$f_i^{kv} \geq \sum_{j \in N} \sum_{h \in N} \sum_{v \in V} f_{jih}^{kv} - M_1(1 - \gamma_i^v) \quad k \in K, i \in N, v \in V \quad (5d)$$

$$f_{jih} \geq x_{ji}^{kv} - x_{ih}^{kv} - M_2(2 - y_{ji}^v - y_{ih}^v + z_{ji}^v + z_{ih}^v) \quad k \in K, j, i, h \in N, v \in V \quad (5e)$$

$$f_{jih} \geq x_{ji}^{kv} - x_{ih}^{kv} - M_3(4 - z_{ji}^v - z_{ih}^v - y_{ji}^v - y_{ih}^v) \quad k \in K, j, i, h \in N, v \in V \quad (5f)$$

$$\sum_{i \in N} \gamma_i^v = 1 \quad v \in V \quad (5g)$$

$$\sum_{(i,j) \in A} z_{ij}^v = 2 \quad v \in V \quad (5h)$$

$$\gamma_i^v - \sum_{j: (i,j) \in A} z_{ij}^v \leq 0 \quad i \in N, v \in V \quad (5i)$$

$$\gamma_i^v - \sum_{j: (j,i) \in A} z_{ji}^v \leq 0 \quad i \in N, v \in V \quad (5j)$$

$$\sum_{j: (i,j) \in A} y_{ij}^v - \sum_{j: (j,i) \in A} y_{ji}^v = 0 \quad i \in N, v \in V \quad (5k)$$

$$\sum_{j: (i,j) \in A} y_{ij}^v - \gamma_i^v \leq 1 \quad i \in N, v \in V \quad (5l)$$

$$e_{ji}^v - e_{ih}^v + M_4(y_{ih}^v + y_{ji}^v - 2 - z_{ji}^v - z_{ih}^v) \leq -1 \quad i, j, h \in N, v \in V \quad (5m)$$

$$y_{ij}^v - \lambda^v \leq 0 \quad (i, j) \in A, v \in V \quad (5n)$$

$$\tau_v \leq t_{max} \quad v \in V \quad (5o)$$

$$\tau_v = \sum_{(i,j) \in A} y_{ij}^v (t_{ij}^v + b_j) \quad v \in V \quad (5p)$$

$$\frac{t_{max}}{\tau_v} u_v y_{ij}^v \geq \sum_{k \in K} x_{ij}^{kv} \quad (i, j) \in A, v \in V \quad (5q)$$

$$z_{ij}^v, y_{ij}^v \in \{0, 1\} \quad (i, j) \in A, v \in V \quad (5r)$$

$$f_{jih}^{kv} \geq 0 \quad k \in K, j, i, h \in N, v \in V \quad (5s)$$

$$f_j^{kv} \geq 0 \quad k \in K, j \in N, v \in V \quad (5t)$$

$$e_{ij}^v \in \mathbb{Z}^+ \quad i, j \in N, v \in V \quad (5u)$$

$$x_{ij}^{kv} \geq 0 \quad (i, j) \in A, k \in K, v \in V \quad (5v)$$

$$\gamma_i^v \in \{0, 1\} \quad i \in N, v \in V \quad (5w)$$

$$\lambda^v \in \{0, 1\} \quad v \in V \quad (5x)$$

$$\tau_v \geq 0 \quad v \in V \quad (5y)$$

The objective function (5a) minimises the total cost of transporting the cargo through the network, the transshipment costs of the demand in butterfly nodes, and the associated cost for deploying the vessels. Furthermore, the flow conservation constraints (5b) ensure that all demand is satisfied. Constraints (5c) account for the amount of containers transshipped in intermediate ports; however, if the corresponding service is non-simple, the model requires constraints (5d)-(5f) for updating the commodities transshipped by the same vessel in butterfly services. Constraints (5g)-(5j) are used to handle butterfly services. Constraints (5g) identify the unique butterfly node for the vessel service and constraints (5h)-(5j) find the adjacent arcs corresponding to the first or last visit to the butterfly node of the vessel service. Moreover, constraints (5k) are the flow conservation constraints for the network design of the vessel service, and constraints (5l) control the number of times a vessel visits the butterfly node in a service. In addition, constraints (5m) use the previous information for correctly enumerating the order in which the vessel traverses the arcs in the vessel service. Additionally, the fleet deployment is controlled by constraints (5n), and the corresponding service length is computed in constraints (5o) and (5p). As first introduced in Agarwal and Ergun (2008), the service length is included in the capacity constraints (5q). However, the model does not require weekly departures for all ports. The inclusion of time for the service in the calculation of the capacity, results in a non-linear model. Therefore, it is necessary to linearise the corresponding constraints (5q) together with (5o)-(5p) in order to obtain a MIP formulation. Finally, constraints (5r)-(5y) define the domains of the decision variables.

The high number of details in the previous model allows the representation of a fairly realistic problem, making it possible to design efficient services, to reduce the overall operational costs. Nonetheless, it can easily be seen that the compact model is computationally hard to solve. The model presents several “big-M” constraints, which produce a very weak relaxation. Techniques such as Branch-and-Bound provide large integrality gaps and poor bounds. The authors propose a Branch-and-Cut algorithm to solve this problem, as the method has presented good results for the VRP and other transportation network design problems. The idea is to solve the previous relaxed problem without the transshipment constraints, (5d)-(5f), and the connectivity constraints (5h)-(5j) and (5m) in butterfly nodes and then, gradually add cuts to the formulation when those constraints are violated. Hence, the authors distinguish between two types of cuts; the connectivity cuts and the transshipment cuts.

The connectivity cuts are inspired by Fischetti et al. (1997) and have been modified to handle butterfly services. Let \mathcal{S} be a subset of ports. Then, the connectivity cuts are defined for each vessel v , for any non-empty subset \mathcal{S} of ports, and for any pair of ports such as $k \in \mathcal{S}$ and $l \notin \mathcal{S}$ as follows:

$$\sum_{i,j \in \mathcal{S}} y_{ij}^v \leq \sum_{h \in N} \sum_{g \in \mathcal{S} \setminus \{k\}} y_{hg}^v + \gamma_k^v - \sum_{e \in N} y_{el}^v + \gamma_l^v + 1 \quad v \in V, \emptyset \subset \mathcal{S} \subset N, k \in \mathcal{S}, l \in N \setminus \mathcal{S} \quad (6)$$

These cuts prevent the service of a given vessel v to be disconnected. Otherwise, if the service is connected, the previous constraints (6) must hold. The cuts ensure that the number of arcs contained in the subset \mathcal{S} has to be smaller or equal to the difference between the sum of the arcs for the vessel service with an end-point in the subset $\mathcal{S} \setminus \{k\}$ and the sum of the arcs in the vessel service with an end-point in the port l , and plus one. Moreover, for butterfly services, the right-hand side of the constraints need to be extended to identify if either port k or l is the butterfly node for the service of vessel v . If a subset \mathcal{S} is identified such as the previous constraint is violated, it means that the service operated by vessel v is not connected and the corresponding cuts are added to the model.

Next, we present the transshipment cuts. As transshipments of containers can only occur in the port identified as the butterfly node of the service, the transshipment cuts are only defined for butterfly services. Given a vessel v , we first identify if the corresponding service is a butterfly service or not. Then, we define $B(A)$ as the set of butterfly services on the set of arcs A , \mathcal{A} as a subset of arcs for a butterfly service in $B(A)$, and $y^v(\mathcal{A})$ as the number of arcs sailed by the corresponding vessel v in the set \mathcal{A} . Moreover, let $T((j,i), \mathcal{A})$ be the simple tour of \mathcal{A} in which the arc (j,i) is located. Hence, the transshipment cut can be constructed as:

$$z_{ji}^v + z_{ih}^v + 2(|\mathcal{A}| - y^v(\mathcal{A})) \geq 2 \quad v \in V, (j, i), (i, h) \in \mathcal{A} \in B(A), T((j, i), \mathcal{A}) \neq T((i, h), \mathcal{A}) \quad (7)$$

The transshipment cuts (7) are defined for each vessel v and pair of arcs in the subset \mathcal{A} not belonging to the same simple tour of \mathcal{A} , as represented by $T((j, i), \mathcal{A}) \neq T((i, h), \mathcal{A})$. The previous cuts make sure that if all arcs in the set \mathcal{A} are sailed in the service operated by vessel v , then the arcs (j, i) and (i, h) belong to different loops. When a set of arcs \mathcal{A} representing a butterfly service is identified violating the previous constraints, the previous transshipment cut is added to the model together with the following constraints:

$$f_i^{kv} \geq \sum_{j, h \in N} \sum_{v \in V} f_{jih}^{kv} - M_1(1 - \gamma_i^v) \quad k \in K, i \in N, v \in V \quad (8)$$

$$f_{jih} \geq x_{ji}^{kv} - x_{ih}^{kv} - M_2(2 - y_{ji}^v - y_{ih}^v + z_{ji}^v + z_{ih}^v) \quad k \in K, j, i, h \in N, v \in V \quad (9)$$

$$f_{jih} \geq x_{ji}^{kv} - x_{ih}^{kv} - M_3(4 - z_{ji}^v - z_{ih}^v - y_{ji}^v - y_{ih}^v) \quad k \in K, j, i, h \in N, v \in V \quad (10)$$

The cuts are found through a separation algorithm based on depth-first search, and the first found violated cut is added to the formulation. The paper presented by Reinhardt and Pisinger (2012) is the first paper studying exact methods for liner shipping with transshipment operations and butterfly services. However, it cannot solve real-life problems from LINER-LIB. Despite this, it gives promising results for small liner shipping network design problems such as feeder services, and computational results are provided for randomly generated instances of up to 15 ports.

3.3 Modelling complex-structures for services

The majority of models for LSNDP are defined using an arc formulation. However, this formulation can be problematic when formulating non-simple services, as it requires the inclusion of many extra variables in the model as seen in Reinhardt and Pisinger (2012). In this section, we briefly present some modelling approaches for modelling non-simple structures for the services.

Port-call formulation for LSNDP

The general idea of this formulation is to define services as a sequence of port-calls. This is done in order to handle non-simple services, because it better reflects how services are designed in practice. By defining these sequences, it is possible to distinguish between multiple calls to the same port during the same service.

Plum et al. (2014b) propose a new mathematical formulation based on a service formulation, where the set of all services S is defined a-priori. This formulation defines service flow variables to model how containers are transported within and between services. The model makes use of a number of continuous flow variables that represent the amount of cargo that is transported on a given leg of a service. Similarly, extra continuous flow variables are defined for accounting for the amount of cargo from and to a port in a given leg of a service. In this way, the service flow can be used to model the transshipment of cargo, and correctly accounting for the transshipment costs. Furthermore, this formulation allows cargo rejection by imposing a penalty in the objective function. The model moreover imposes the services to have weekly frequency, while limiting the fleet deployment according to the available fleet. Finally, the authors define the objective function to maximize the profit of the transported cargo, while minimising the operational cost of the services and handling costs of cargo.

The problem is proved to be *NP*-hard by reduction from the TSP, and the authors report promising solutions for the two smallest LINER-LIB instances using CPLEX. However, due to the large number of constraints and decision variables, the optimal solutions are not achieved.

3.4 Outbound-inbound principle

The geographical distribution of the ports and the coastal lands may limit the structure of the services. In the case where the ports follow a natural sequence, it may be convenient to define services following a so-called “outbound-inbound principle”, i.e. each service goes back and forth along this natural sequence of the ports. These services are similar to pendulum services; however, they may be asymmetric, as some ports can be omitted in each direction, as illustrated in Figure 8.

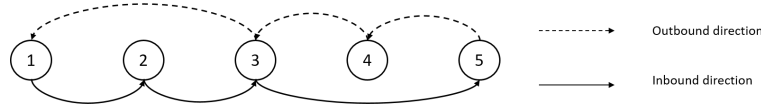


Fig. 8. Example of a asymmetric service with five ports with the outbound-inbound principle.

Wang and Meng (2014) study the LSNDP with services following this structure and incorporate transit time constraints. However, transshipments between services are excluded in this approach, as they are less significant when working with this kind of services. The proposed model is mixed-integer, non-linear and non-convex and it both designs the service network, as well as determines the cargo flow through the network, while considering a limited fleet.

The demand is allowed to be split between different services, and the model uses continuous variables, representing the amount of containers flowing through the arcs. Moreover, the model enforces maximum transit times, between each pair of nodes, and containers cannot be served by a service which uses more than this maximum time, to get from the containers origin to their destination. Finally, the model defines the port times as a functions of the number of containers handled at the ports and the services are enforced to have weekly frequencies.

The problem is proved to be strongly *NP*-hard by reduction from the Bin Packing Problem, and to handle the non-linearities as well as making the problem more tractable, Wang and Meng (2014) describe a column generation based algorithmic scheme to solve it. The approach efficiently finds high-quality solutions that can help planners to design better liner shipping networks. The algorithm is tested on instances consisting of 12 ports and 3 vessels classes. However, the comparison with exact solutions is studied with instances having 7 ports, as CPLEX is able to find the optimal solution in these cases.

3.5 Layer-network for complex services structures

Thun et al. (2017) propose a new mathematical formulation for the LSNDP where all kinds of service-structures are allowed, i.e. each port in the service can be visited several times. The authors present an approach where the generation of services is separated from the fleet deployment and the coordination of services, which is carried out following a Column Generation approach. The innovative feature about this modeling is given by the way in which the network is defined for creating the services.

The master problem is responsible for coordinating the services, which are known, and a service is given as a sequence of ports, a vessel type and a number of assigned vessels. Additionally, the set of possible delivery patterns for each service is defined, where information about the amount of cargo loaded and unloaded in the ports is provided. However, this set can be very large, and the model allows any convex combination of generated delivery patterns, as long as they add up to an integer number of delivery patterns for each service. The master problem is defined with decision variables for the selection of a service with a demand pattern, subject to the constraints that all demand must be transported, respecting the availability of the fleet, while minimising the cost associated to the selected services.

As mentioned above, the generation of the entire set of services can be quite time-consuming. Therefore, Thun et al. (2017) propose a sub-problem for each vessel type for finding new services and delivery patterns.

To allow complex route structures, the services are modelled using a multi-layer graph, where each layer is a complete sub-graph containing exactly one copy of each port. The layers are then connected with edges between the nodes representing the same port. This way, a port can be visited multiple times, but in different layers, while keeping the benefits from working with simple routes in the extended network. A similar approach, to use a multi-layered graph to model complex services, was also earlier used by Guericke and Tierney (2015). A graphical representation of the network can be seen in Figure 9. Using this network, the sub-problem is then to create the services with minimum reduced cost, satisfying constraints on speed and sailing time. An arc formulation is used to model these sub-problems, where the flow of the cargo is controlled with continuous variables, taking into account the amount of transshipped cargo from other services.

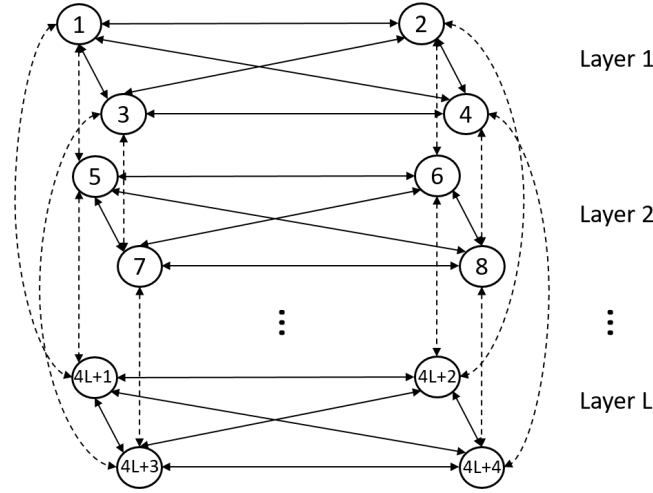


Fig. 9. Visual representation of the layer-network with four ports and L layers.

It can easily be observed that the complexity of the sub-problem increases with the number of layers. Thun et al. (2017) impose a limitation of two layers in the defined graph. This is a realistic limitation, as a service seldom visit a port more than twice. Hence, the sub-problem allows the creation of both simple, butterfly and pendulum services. The problem is solved to integer optimality using Branch-and-Price.

In each node of the tree, the feasibility of the solution is verified with respect to the integrality restriction of the convex combination of the delivery patterns. If integrality does not hold, the tree is branched by stages. In the first stage of the branching, an integer number of assigned vessels is searched for. Once it is obtained, the second stage similarly branches in the number of visits to each port for each of the vessel classes. Finally, in the third stage, the branching is applied on the total number of time the vessels of a certain type sails between two ports. Although this branching is usually sufficient, fractional solutions can still be obtained with respect to the number of vessels deployed on a service. Consequently, a second stage arc-branching can be applied, branching on the total number of times vessels of a certain class sails between ports on two consecutive voyages.

The algorithm is tested in small instances of between 5 and 7 ports, presenting between 7 and 14 demands, and is compared with cases where the structure of the services is limited. The authors present results showing that when considering more complex service structures, solutions with better cost-effective networks can be obtained.

4 Two-stage algorithms

The LSNDP consists of two tightly interrelated problems — the vessel service network design and the container flow problem. One of the most successful approaches so far for finding good solutions to the LSNDP, has been to use heuristics exploiting this two-tier structure.

The idea, in general, is to first generate a set of services for the vessels and then to solve the container flow problem, given the set of services. It is then common to use information from the container flow to update the services. This way a feedback loop is created, iteratively improving the services and solving the container flow. The different frameworks, in which this has been used, range from column generation and Bender's decomposition (Agarwal and Ergun (2008)) to various matheuristics (Álvarez (2009), Brouer et al. (2014b)). This section will discuss some of those methods. Various versions of the LSNDP will be featured, both with and without transshipment costs, transit time constraints and rejection of demand.

4.1 The container flow problem

Before going into the full two-stage algorithms, let us briefly discuss the container flow problem, which is the lower tier problem in the LSNDP two-tier structure. In general, for a given set of services, the container flow problem reduces to a *multi-commodity flow* problem (MCFP) with fractional flows allowed.

In addition to the notation defined in Section 1.2, we need an additional parameter: to each arc $(i, j) \in A$ define its corresponding flow capacity, u_{ij} . The arc set A and its corresponding costs c_{ij}^k and capacities u_{ij} are defined by the vessel services, designed in the upper-tier problem. Also, let x_{ij}^k be a continuous variable denoting the flow of commodity k through arc (i, j) . The MCFP can then be expressed as:

$$\min \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k \quad (11a)$$

$$\text{s.t.} \quad \sum_{j \in N: (i,j) \in A} x_{ij}^k - \sum_{j \in N: (j,i) \in A} x_{ji}^k = \xi_i^k \quad i \in N, k \in K \quad (11b)$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij} \quad (i, j) \in A \quad (11c)$$

$$x_{ij}^k \geq 0 \quad (i, j) \in A, k \in K. \quad (11d)$$

Here, the objective, (11a), is to minimise the total cost. Constraints (11b) are the flow conservation constraints, constraints (11c) are the capacity constraints, and constraints (11d) define the domain of the variables x_{ij}^k .

When fractional flows are allowed, the MCFP is solvable in polynomial time. For larger instances, it is, however, still computationally demanding. As the model generally has to be solved a multitude of times in the presented two-tier solutions to the LSNDP, efficient solution methods to the MCFP are essential.

One of the most common solution approaches is to exploit its block-angular constraints matrix and apply Dantzig-Wolfe Decomposition (Ahuja et al. (1993); Karsten et al. (2015)). The problem should first be reformulated as a path-flow formulation, where the goal is to allocate the commodities to a number of flow paths from the commodities' origins to their destinations, while respecting the capacity constraints on the arcs. Let P^k be the set of all paths for commodity $k \in K$, from o_k to d_k , and let P_a^k be the set of paths for commodity k , which uses arc a . Then we define

$$P_a = \bigcup_{k \in K} P_a^k,$$

to be the set of all paths going through arc $a \in A$. For each path, p , for commodity $k \in K$, define its cost $c_p^k = \sum_{a \in A: p \in P_a^k} c_a^k$, and a corresponding decision variable f_p^k , deciding the flow through path p . The path-flow formulation can then be expressed as:

$$\min \sum_{k \in K} \sum_{p \in P^k} c_p^k f_p^k \quad (12a)$$

$$\text{s.t.} \quad \sum_{p \in P^k} f_p^k = \xi_{o_k}^k \quad k \in K \quad (12b)$$

$$\sum_{p \in P_a} f_p^k \leq u_a \quad a \in A \quad (12c)$$

$$f_p^k \geq 0 \quad k \in K, p \in P^k. \quad (12d)$$

The objective function, (12a), minimises the cost. Constraints (12b) ensure that all commodities are delivered and constraints (12c) assert that the arc capacity cannot be exceeded. Lastly, constraints (12d) define the domain for the variables.

The path formulation has a very large number of variables, but generally, only a few of them are needed for the optimal solution. Using column generation, the problem can be restricted to only consider a limited amount of paths for each commodity and new paths can then be generated dynamically. In this way, the path formulation can generally be solved faster than the arc formulation, described earlier. The path formulation makes it relatively easy to implement transit time constraints as they can be handled in the pricing problem.

4.2 Matheuristics methods for the LSNDP

While the lower-tier container flow problem is solvable in polynomial time (when no transit time constraints are imposed), the upper-tier service selection problem is *NP*-hard, and just calculating the objective value of a given solution, demands solving the container flow problem. This makes the service selection problem difficult to solve to optimality and instead several matheuristics have been developed to find good solutions to larger instances. A matheuristic is a method that employs heuristics together with methods from linear and integer programming. In the case of the LSNDP, the most common procedure is to use linear programming tools to solve the MCFP and then various heuristics to update the vessel services.

The first two-stage algorithms for liner shipping network design were presented by Agarwal and Ergun (2008), that solved the *simultaneous ship scheduling and cargo routing problem* (SSSCRCP) with a column generation and a Bender's decomposition heuristic. As the name implies, they also took the ship scheduling into account which has been more or less neglected since. They did not, however, account for transshipment costs. The column generation heuristic was designed such that the cargo routing was solved in the master problem, and the dual variables were then utilised to generate and choose new services for the vessels. Once no more services with negative reduced cost could be found, they used the generated columns to find an integer solution using branch-and-bound. In the Bender's decomposition heuristic, the container flow problem was solved in the subproblem to add optimality cuts for the service generation in the master problem. In both cases they found it most efficient to generate new services using a labelling algorithm. They reported good results for instances of up to 20 ports and 100 vessels.

Another prominent approach was presented by Álvarez (2009), that used a matheuristic which perturbed the services with a tabu-search scheme, solved the container flow problem using an interior point method and generated new services from the dual variables from the container flow solutions. Álvarez's model included the cost of transshipments and also allowed for butterfly services. The moves considered in the tabu-search for the services are deletion, change in vessel speed and change in number of vessels assigned. To guide the search, from the solution of the commodity flow, he checked which services

are under/over utilised and added/removed vessels and increased/decreased speed, where necessary. The paper presents computational results for up to 100 available vessels and 120 ports.

Another tabu-search approach was presented by Brouer et al. (2015) and was later improved upon by Karsten et al. (2017b), by adding time-constraints for the commodities. As it is computationally costly to solve the full cargo flow problem, both papers instead developed a method to estimate the impact of a change in the service structure. Their solution method is then based on an improvement heuristic, first presented by Archetti and Speranza (2014), where an integer program is solved to update the current services, in each iteration.

Here follows a brief description of the algorithm from Brouer et al. (2015). The algorithm is initialised by using a greedy knapsack heuristic to generate an initial set of services. The change in revenue and time by including or excluding ports from the current services is estimated by solving a set of shortest path problems. In addition to the notation from the Section 1.2, let τ_s be the time length of a service s , let Δ_{is}^{R+} (Δ_{is}^{R-}) be the estimated revenue change and Δ_{is}^{T+} (Δ_{is}^{T-}) be the estimated duration change from including (excluding) port $i \in N$ in (from) service $s \in S$. Also, let η_s^+ (η_s^-) be the maximum number of inclusions (removals) allowed and let \bar{N}_s denote the set of ports which can be included. Let \hat{m}_v denote the number of free vessels of class v , such that

$$\hat{m}_v = m_v - \sum_{s \in S} m_v^s.$$

Lastly, let us define the binary variables x_{is}^+ and x_{is}^- , which control the inclusion and removal, respectively, of port i from service s , and the integer variables ζ_s , which denote the number of vessels to add to/subtract from service s . For each service $s \in S$, with corresponding vessel class v , we can then define the following integer program:

$$\max \sum_{i \in N_s} \Delta_{is}^{R+} x_{is}^+ + \sum_{i \in \bar{N}_s} \Delta_{is}^{R-} x_{is}^- - c_v \zeta_s \quad (13a)$$

$$\text{s.t. } \tau_s + \sum_{i \in N_s} \Delta_{is}^{T+} x_{is}^+ + \sum_{i \in \bar{N}_s} \Delta_{is}^{T-} x_{is}^- \leq 24 \cdot 7 \cdot (m_v^s + \zeta_s) \quad (13b)$$

$$\zeta_s \leq \hat{m}_v \quad (13c)$$

$$\sum_{i \in \bar{N}_s} x_{is}^+ \leq \eta_s^+ \quad (13d)$$

$$\sum_{i \in N_s} x_{is}^- \leq \eta_s^- \quad (13e)$$

$$\sum_{j \in L_i} x_{js}^- \leq |L_i| (1 - x_{is}^+) \quad i \in \bar{N}_s \quad (13f)$$

$$\sum_{j \in L_i} x_{js}^- \leq |L_i| (1 - x_{is}^-) \quad i \in N_s \quad (13g)$$

$$x_{is}^+ \in \{0, 1\}, \quad i \in \bar{N}_s \quad x_{is}^- \in \{0, 1\}, \quad i \in N_s \quad \zeta_s \in \mathbb{Z}, \quad (13h)$$

Here, the objective (13a) is to maximise the increase in revenue. Constraint (13b) ensures that there is enough vessels assigned to keep the weekly frequency, and constraint (13c) says that no more than the number of free vessels can be added to the service. Constraints (13d) and (13e) set a limit on the number of insertions and removals, while (13f) and (13g) prevent certain combinations of insertions and removals. The sets L_i^+ are defined such that if a port i is to be inserted, then no port in L_i^+ is allowed to be removed. If instead a port i is removed, then every port in L_i^- must remain. If a new port call is inserted in between two ports, then neither of those are allowed to be removed, and if inserting a new port means that a new commodity is transported, then the origin and destination nodes, of this commodity, are not

allowed to be removed. Constraints (13d)–(13g) are defined to limit the amount of changes which can be applied, as the revenue and time change estimates are made for one or a few changes and deteriorates rapidly when multiple changes are applied. L_i are defined such that if a port i is to be inserted in between two ports, then neither of those are allowed to be removed, and if inserting a new port means that a new commodity is transported, then the origin and destination nodes, of this commodity, are not allowed to be removed. Lastly, (13h) define the domain of the variables.

The algorithm works such that each service, one by one, is updated according to the solution of the above defined mixed integer problem. Then the MCFP is solved to update the total revenue, and the effect of new changes is once again estimated with the shortest path procedure. To diversify the solutions, in every tenth iteration the services with lowest utilisation are removed and new services are created using the greedy creation heuristic.

Brouer et al. (2015) report satisfactory solutions for 6 out of 7 instances from the LINER-LIB benchmark set where the largest solved instance, the *world small*, contains 47 ports and 317 available vessels.

5 Subset of Routes

Balakrishnan and Karsten (2017) suggest a method for generating a network by selecting a subset of sailing services from an initial pool of candidate services given by expert planners in advance. The problem is therefore reduced from service design to service selection. Limits on the number of transshipments for each container are included in the model and rejection of demand is allowed. This profit maximising problem is denoted the Liner Service Planning (LSP) problem.

We extend the basic notation from Section 1.2 to include A_s as the set of sailing arcs associated with each candidate service $s \in S$. Each arc $a \in A_s$ represents the part of a ship's itinerary between two successive ports on the service route. Associated with each service $s \in S$ is also a cost c_s and for each arc $a \in A_s$ a capacity u_a . It is allowed to split the flow of each commodity and a penalty cost c_R^k per container is used to penalise rejected demand of commodity k .

Given a commodity's route, a *sub-path* is defined as the part of the route in which the containers travel on a single service. If this part is from port i to port j on service s , the sub-path is denoted $\langle i, j, s \rangle$. The set H_s denotes the full set of sub-paths for service s , i.e. the set contains one sub-path $\langle i, j, s \rangle$ for each combination of ports i and j included in service s . These sub-paths are used to introduce an augmented multi-commodity flow network in order to incorporate the limits on the number of transshipments and their associated costs. This modelling approach falls somewhere between the two more traditional modelling approaches of either using arc-flow, i.e. flow over sailing edges, or path-flows, i.e. flow over origin-to-destination paths.

The augmented network contains one node for each port and one link for each sub-path of each service. The sub-path structure also extends to more complex routes, e.g., butterfly routes. A_{ij}^s denote the set of sailing arcs of service s included in sub-path $\langle i, j, s \rangle$. The cost of routing one container of commodity k on sub-path $\langle i, j, s \rangle$ is denoted c_{ijs}^k . Finally, r_k denote the maximum allowed number of sub-paths on which commodity k can travel. Note that r_k must be one larger than the maximum permitted number of transshipments to enforce this constraint.

Balakrishnan and Karsten (2017) present a multi-commodity model based on flows along sub-paths in the augmented network. The binary variable y_s is equal to 1 if service $s \in S$ is selected, and 0 otherwise. The flow of commodity k using sub-path $\langle i, j, s \rangle$ as the h^{th} stage is defined by the variable x_{ijs}^{hk} for $s \in S$, $\langle i, j, s \rangle \in A_s$, and $h = 1, 2, \dots, r_k$. Finally, z_k is equal to the unmet demand (number of containers) for commodity $k \in K$.

The LSP problem can then be described by the following mixed-integer program:

$$\min \sum_{s \in S} c_s y_s + \sum_{k \in K} \sum_{s \in S} \sum_{\langle i, j, s \rangle \in A_s} \sum_{h=1}^{r_k} c_{ijs}^k x_{ijs}^{hk} + \sum_{k \in K} c_k^R z_k \quad (14a)$$

$$\text{s.t. } \sum_{s \in S} \sum_{\langle o_k, j, s \rangle \in H_s} x_{o_k j s}^{1k} + z_k = q_k \quad \forall k \in K, \quad (14b)$$

$$\sum_{h=1}^{r_k} \sum_{s \in S} \sum_{\langle j, d_k, s \rangle \in H_s} x_{j d_k s}^{hk} + z_k = q_k \quad \forall k \in K, \quad (14c)$$

$$\sum_{s \in S} \sum_{i: \langle i, j, s \rangle \in H_s} x_{i j s}^{hk} - \sum_{s \in S} \sum_{l: \langle j, l, s \rangle \in H_s} x_{j l s}^{h+1, k} = 0 \quad \forall k \in K, j \in N \setminus \{o_k, d_k\}, h = 1, \dots, r_k - 1, \quad (14d)$$

$$\sum_{k \in K} \sum_{h=1}^{r_k} \sum_{\langle i, j, s \rangle \in H_s, a \in A_{ij}^s} x_{i j s}^{hk} \leq u_a y_s \quad \forall s \in S, a \in A_s \quad (14e)$$

$$\sum_{s \in S} m_v^s y_s \leq m_v \quad \forall v \in V, \quad (14f)$$

$$x_{i j s}^{hk} \geq 0 \quad \forall k \in K, s \in S, \langle i, j, s \rangle \in H_s, h = 1, \dots, r_k, \quad (14g)$$

$$z_k \geq 0 \quad \forall k \in K, \quad (14h)$$

$$y_s \in \{0, 1\} \quad \forall s \in S. \quad (14i)$$

The objective function (14a) minimises total cost comprised of fixed costs for the selected services, the cost of transporting commodities along each sub-path, and finally the penalties incurred for rejected demand. By including penalties the problem is formulated as a cost minimisation problem as opposed to a profit maximisation problem where c_R^k would instead represent the revenue for transporting one unit of commodity k .

Constraints (14b) and (14c) ensure that the flow of each commodity k is assigned to sub-paths incident to the corresponding origin port o_k and the destination port d_k . They also ensure that the flow out of the origin port in combination with the unmet demand for commodity k adds up to the total demand for commodity k . Constraints (14d) are flow-balancing constraints for intermediate ports. Together with constraints (14b) and (14c), these constraints ensure that for each commodity k , the demand, minus any unmet demand, will arrive at the destination port using at most r_k sub-paths, i.e. fulfilling the constraint on a maximum number of transshipments.

Constraints (14e) impose capacity constraints on the sailing arcs and ensure that only sub-paths from the selected services are used. Constraints (14f) ensure that no more than the available vessels are used. Finally, constraints (14g)-(14i) impose non-negativity and binary restrictions on the respective decision variables.

The LSP model formulation is flexible enough to allow incorporation of several practical container routing issues such as cabotage rules, regional policies and embargoes. The incorporation of many of these constraints can be handled during preprocessing simply by removing sub-paths that are no longer permitted.

Balakrishnan and Karsten (2017) show that the LSP problem is *NP*-hard. A problem reduction procedure to eliminate or combine variables is outlined and valid inequalities for increasing the lower bounds of its linear programming (LP) relaxation are described.

5.1 Optimisation-based Heuristic Procedure

Balakrishnan and Karsten (2017) propose an optimisation-based heuristic algorithm to generate good initial solutions. The heuristic iteratively solves the LP-relaxation of the problem and fixes service selection variables, y_s , that are integer in the corresponding solution, and rounds service selection variables, y_s , that are fractional. The highest or lowest fractional variable is selected in each iteration and rounded up or down correspondingly. The heuristic procedure first rounds down low y -values before rounding up high y -values. Thereby, unattractive services are eliminated early in the process. If rounding a variable up

causes a violation of the fleet availability constraints, the variable is instead set to zero. The LP-relaxation is then re-solved. When all y_s variables assume binary values the procedure stops.

Balakrishnan and Karsten (2017) test their solution method on four data sets from the LINER-LIB benchmark suite with at most two transshipments per container. The initial pool of candidate services is generated using the matheuristic from Brouer et al. (2014b). The LP-based heuristic yields solutions that are close to optimality in relatively short time. This method can therefore be used as a stand-alone tool or to warm-start an exact solution procedure.

6 Backbone flow

The main idea in the backbone flow algorithm, as presented by Krogsgaard et al. (2018), is to reverse the order of two-phase algorithms by first flowing the containers, and then constructing services that cover the flow.

The backbone flow algorithm uses a complete and directed graph $G = (N, A)$. There are no capacities associated with the edges, but the cost of using an edge (i, j) depends on how many containers in total are flowing on that edge. This can be expressed as a concave function $c(x)$ of the flow x reflecting the economy of scale for flowing more containers: There is a large cost associated with opening an arc (i.e. deploying a vessel), while the cost per container decreases as the flow (and hence vessel size) is increased. See Figure 1 for an illustration of the costs. The cost function implicitly aims at aggregating the flow on fewer arcs. Sun and Zheng (2016) also use a concave function to optimise the container flow.

Let x_{ij}^k denote the flow of commodity k on edge (i, j) . Then the backbone flow problem becomes a non-linear MCFP as given by

$$\min \sum_{(i,j) \in A} c\left(\sum_{k \in K} x_{ij}^k\right) \quad (15a)$$

$$\text{s.t.} \quad \sum_{(i,j) \in A} x_{ij}^k - \sum_{(j,i) \in A} x_{ji}^k = \xi_i^k \quad i \in N, k \in K \quad (15b)$$

$$x_{ij}^k \geq 0 \quad (i, j) \in A, k \in K. \quad (15c)$$

As before, the objective, (15a), is to minimise the total cost, and constraints (15b) are the flow conservation constraints. Constraints (15c) define the domain of the variables.

Since the model is non-linear, Krogsgaard et al. (2018) solve the problem heuristically through a randomised greedy algorithm. As the arc costs depend on previously flowed containers, the result of the flow will be very dependent on the order in which containers are flowed. Generally, the first containers are more decisive for the arcs used heavily in the final solution than the last containers flowed. It is thus necessary to run several iterations of the problem, with a random order of the containers, to achieve a reasonable *average* picture of the backbone flow. Running ten iterations for the demand matrix of the *WorldSmall* instance gives the average arc loads shown in Figure 10. The figure clearly shows that only a fraction of the possible arcs is used in the solution.

6.1 From backbone flow to network design

Having found a backbone flow, Krogsgaard et al. (2018) present a greedy heuristic for generating services. The idea is to add one arc at a time to a service until all services have reached their maximum duration.

To generate a service, the unserved arc with the largest flow is selected as the first arc in the service, and a return arc is added to close the service. While the service is at or below the desired duration, a new arc is added to the service to expand it, and this arc replaces the return arc. The new arc is the unserved

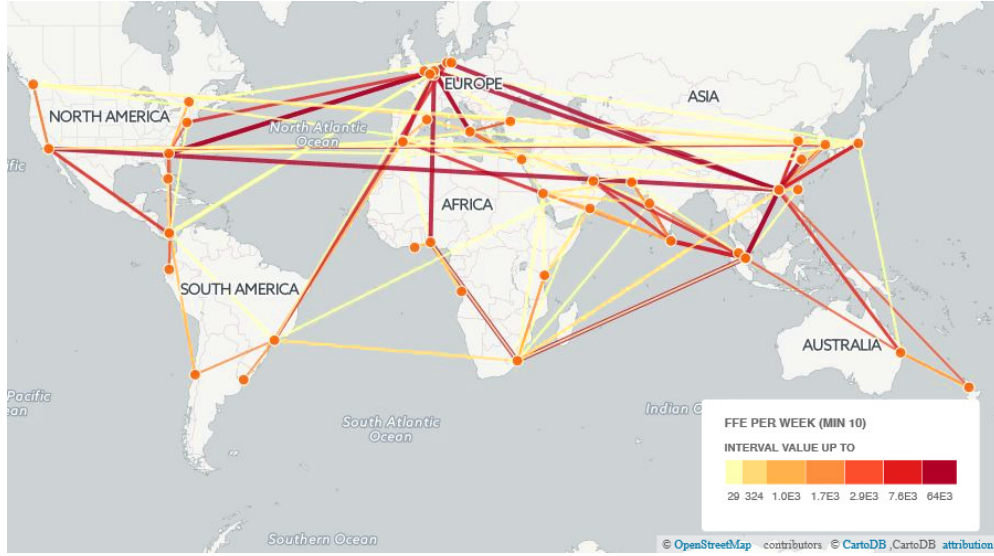


Fig. 10. Typical backbone flow for the *WorldSmall* instance. Source: Krogsgaard et al. (2018).

arc with the largest demand that either starts at the same port as the return arc, which is to be replaced, or ends at the same port as the return arc. A new return arc is added to close the service. The selection process continues until it is not possible to add a new arc without exceeding the maximum duration of the service. After this, the creation of the next service starts.

To obtain a number of different start solutions to select from, the algorithm is repeated a number of times with random settings on the maximum service length for every service. The length is selected in a predefined interval depending on the size of the vessel, such that larger vessels, typically travelling between continents, get longer services than smaller vessels doing feeder service. For every service generated, a duration is selected in the interval at random, and the service is constructed. This is repeated until all available resources have been exhausted.

In the computational study by Krogsgaard et al. (2018) it is shown that usable solutions can be found in relatively short time. Using the *WorldSmall* instance, the authors generate 20 different sets of services by running the above algorithm where the containers are flown in random order. This can be done in about 80 seconds, and results in profitable network, although the resulting network is far from optimal.

In order to improve the initial services found by the greedy heuristic, Krogsgaard et al. (2018) use a Variable Neighbourhood Search (VNS) algorithm to reach a high quality network. The general idea in VNS, as presented by Hansen and Mladenovic (2014), is to apply different neighbourhood structures throughout the search to exploit the benefits from neighbourhood changes. Promising computational results using the LINER-LIB instances are reported as will be described in Section 7. Perhaps the most important result is that the number of transshipments in general is very low, being around 1.14 on average per commodity. For the smaller instances, the number of transshipments is around 0.5 per commodity. Fewer transshipments means shorter port stays, and hence vessels can sail at slower speed between the ports.

7 Computational results for LINER-LIB instances

In this section we compare computational results for most of the presented models, using the LINER-LIB instances introduced in Section 1.4. Table 3 reports solution values and computational times for

LINER-LIB instances *without* transit time limits, while Table 4 reports the same values for instances *with* transit time limits. Objective values are reported in k\$, while CPU-time is reported in seconds.

The objective function minimises the operational costs, hence negative values indicate a profitable network. Also, we indicate whether the authors operate with weekly or bi-weekly frequency, and whether constant speed or variable speed is used. For the instances without time constraints, constant speed is always used.

Balakrishnan and Karsten (2017) only report relative improvements in their paper, hence the objective values have been obtained from the authors. Krogsgaard et al. (2018) maximize profit instead of minimising operational costs, hence the objective values have been converted to the minimisation form. Krogsgaard et al. (2018) report two different objective values. The first value, $obj.^h$, is the objective value obtained by solving the container flow problem from Section 4.1 heuristically. The second value, $obj.^c$, is the objective value obtained by solving the flow problem to optimality with CPLEX, using the same network. The flow problem of *WorldLarge* could not be solved by CPLEX. Koza et al. (2019) consider an integrated LSND and scheduling problem, hence transshipment times depend on the exact schedule. In order to compare the results to other algorithms, we report the results from Koza et al. (2019) where the same (48 hours) transshipment times are used as in all the other algorithms in this section.

Looking at the results in Table 3 for instances *without* transit time limits, it is seen that Brouer et al. (2014a) generally obtains good results for the small and medium-sized instances, but the algorithm is using bi-weekly frequency at some services, making it possible to utilise the vessels in a better way than the other algorithms. If weekly operation is requested, then Brouer et al. (2014b) and Krogsgaard et al. (2018) generally have the best performance. Brouer et al. (2014b) reports the best results for *Baltic* and *AsiaEurope* while Krogsgaard et al. (2018) reports the best results for *WestAfrica*, *Pacific*, *WorldSmall* and *WorldLarge*. The CPU times are also very fast, with only 1 hour for solving even the biggest instances.

For instances *with* transit time limits, reported in Table 4, it is seen that Koza et al. (2019) obtain some very impressive results for the small instances, but solution times are very large. For larger instances Karsten et al. (2017a) show good results with reasonable CPU times.

Looking at the flow values, generally all algorithms are able to service 85–98% of the containers. It is generally more difficult to flow all containers for the large-sized instances, in particular if time constraints are imposed. Since the demands in LINER-LIB have been constructed by merging networks and demands from several companies it is not necessarily possible to fulfill all demands, nor to make a profitable network.

	Brouer et al. (2014a)	Brouer et al. (2014b)	Balakrishnan and Karsten (2017)	Krogsgaard et al. (2018)		
	weekly/bi-weekly freq.	weekly freq.	weekly freq.	weekly freq.		
	obj. flow time	obj. flow time	obj. flow time	obj. ^h flow time	obj. ^c	
Baltic	-8,365 95.0 300	-6,160 92.1 105	98 87.9 0	-2,314 92 30	-2,314	
WestAfrica	-143,110 95.3 900	-140,000 97.0 93	-5,039 96.9 2	-144,514 91 90	-144,514	
Mediterranean	12,209 98.8 1 200	32,200 93.8 319	1,837 90.5 237	51,943 83 120	50,140	
Pacific	-54,087 96.9 3,600	69,800 95.7 3,603	3,031 98.1 22,633	-47,057 88 360	-55,030	
AsiaEurope	-657,972 89.6 14,400	-817,000 93.8 14,517	– – –	-668,314 85 1,440	-729,260	
WorldSmall	-1,152,761 88.2 10,800	-1,400,000 94.2 10,885	– – –	-1,304,229 83 1,080	-1,495,030	
WorldLarge	– – –	– – –	– – –	-653,400 82 3,600	–	
	(best of 10)	(best of 10)		(best of 10, 20, 5)		

Table 3. Results for LINER-LIB instances *without* transit time limits. Krogsgaard et al. report their best result out of 10 for *Baltic*, *WestAfrica* and *AsiaEurope*, out of 20 for *Mediterranean*, *Pacific*, *WorldSmall* and out of 5 for *WorldLarge*.

	Brouer et al. (2015)			Karsten et al. (2017b)			Karsten et al. (2017a)			Koza et al. (2019)		
	Constant speed			Constant speed			Variable speed			Variable speed		
	obj.	flow	time	obj.	flow	time	obj.	flow	time	obj.	flow	time
Baltic	-14	87.4	101	-14	87.4	101	-5	87.9	144	-284	92.1	900
WestAfrica	-5,590	97.0	255	-5,590	97.0	255	-5,480	97.6	362	-6,000	96.7	3,600
Mediterranean	2,420	86.9	710	2,420	86.9	710	2,190	83.8	1,200	2,080	81.8	14,400
Pacific	3,050	93.3	3,600	3,810	94.7	3,600	1,130	90.3	3,600	-940	91.1	28,800
AsiaEurope	-16,700	88.8	14,400	-17,600	90.7	14,400	-18,800	85.6	14,400	-20,700	81.4	129,600
WorldSmall	-35,400	91.1	10,800	-31,800	90.7	10,800	-40,500	89.1	10,800	-53,200	88.5	43,200
WorldLarge	—	—	—	—	—	—	—	—	—	—	—	—
	(best of 10)			(best of 10)			(best of 10)			(best of 12)		

Table 4. Results for LINER-LIB instances *with* transit time limits. No results have been reported for the WorldLarge instances.

8 Concluding remarks and future challenges

Liner shipping is the backbone of international trade, and hence it is important to develop decision support tools that can help design more cost-efficient services, and balance several objectives. This includes finding the right trade-off between speed, transportation times, number of transshipments, and operational costs.

Although liner shipping generally is one of the most energy-efficient modes of transportation per kilometer, the shipping industry emits large quantities of SO_x and NO_x. The International Maritime Organization (IMO) has stated the goal that the CO₂ emission from maritime operations should be reduced by 50% until 2050. The stricter limits on the amount of emitted pollutants by container vessels mean that speed optimisation will play a more important role while designing the shipping network, as reducing the vessel speed goes hand in hand with reducing CO₂ emissions. However, this restriction does not only affect the operational speed of the services, but also the overall liner shipping network design. Some port connections may be unreachable by some vessel classes when imposing emission control regulations in certain areas, forcing the vessels to take alternative routes. This also affects the cargo routes, as the transit time constraints for the cargo must still be respected. Hence, better liner shipping network design and optimisation of the affiliated activities may be an important tool in reaching the goal, though this will involve both industrial developments and algorithmic newthinking.

8.1 Industrial development and future research areas

Slow steaming together with larger vessels has proven to be an efficient tool for reducing the energy consumption. However, slow steaming decreases the capacity of vessels, since they cannot transport as much cargo per time unit as before. Hence, more vessels are needed in order to maintain the same capacity, straining the environment. Furthermore, for perishable goods or high-value cargo, longer transportation times might not be an option. We may therefore see in the future that we will have a high-speed network for perishable and high-value cargo, and a low-speed network for the remaining cargo. The two networks need to collaborate since they share the feeder network, and perishable goods may use the low-speed network for shorter distances while the non-perishable goods may use the high-speed network if the price is competitive. So network design of the high- and low-speed network should be addressed together to ensure a good utilization.

Larger vessels tend to be more energy efficient per container, but the increased capacity results in longer port stays, making it necessary to speed up between ports. It is therefore necessary to design services such that fewer port calls are needed, while still ensuring a good utilisation of the mega vessels.

By having fewer port calls, we save the pilot and mooring time in the ports, and hence can sail slower at the long distances. This corresponds to the development in the airline industry, where the biggest airplanes only operate direct flights between the biggest hubs.

Moreover, in the future, we will see container vessels operating with new, greener, propulsion methods. Electric vessels may operate shorter services, while liquid natural gas (LNG) may be used for operating longer services. **The new propulsion types will most likely radically change how service networks should be designed**, since refueling/recharging will be more complicated, and vessels will have a more limited range of operations.

The introduction of autonomous vessels in the container shipping industry may also significantly change the way a network is designed and operated. In particular for feeder-lines we may see more smaller vessels sailing on-demand, depending on the cargo. This would turn the network design process into a dynamic routing problem. It would also be interesting to investigate whether fuel savings are achievable if autonomous vessels are sailing in convoys close to each other. If this is the case, the network design should take these convoys into account.

Nearly every vessel will be delayed in one or more ports during a round trip. Instead of just speeding up (and hence using more energy) advanced disruption management tools need to be developed that can ensure timely arrival to the end customer with the lowest possible energy consumption. **Some studies along this path include Brouer et al. (2013), who proposed a MIP model for handling disruptions in liner shipping; and Wang and Meng (2012), who included uncertain wait and handling times at ports for designing a more robust shipping network.** However, much more work needs to be done in this area.

Vessel sharing agreements are an important tool for making it possible to operate larger and more energy-efficient vessels. In a vessel sharing agreement, two or more companies share the capacity of a vessel throughout the full rotation or on certain legs. Vessel sharing agreements, however, substantially increase the complexity of designing a network, since some legs and capacities are locked according to the agreement.

8.2 Industrial and academic synergy

Shipping companies currently use the experience of decision makers and planners to carry out most of the logistic and planning decisions. To aid this decision process with high-level analytical support-tools would have a massive impact, both from an economical, as well as an environmental, point-of-view. However, optimisation in liner shipping is still in early research stages, and the implementation of the current designed models and for algorithms to be used in real-life applications is one of the main future challenges.

During recent years, there has been an increase in the literature on optimisation in liner shipping. Nonetheless, the majority of the assumptions may be unrealistic and most approaches so far are rather abstract and do not capture all of the practical constraints and restrictions faced by the liner shipping industry. Therefore, a strong collaboration between industry and academia is necessary, in order to develop more realistic decision-support tools which can be used by liner shipping companies in real-life applications. However, this is a challenging task, as it requires the detailed implementation of many complex business rules and considerations, some of which are most likely not yet completely understood or considered in the literature.

Finally, there is also a significant challenge from the perspective of change management – to reach out to the industry and earn their trust, so that they will adopt those new methodologies, once they are ready. It is key that the developed tools should be made understandable for planners, in order to encourage and foster their use by the shipping companies.

8.3 Algorithmic challenges and directions

It should be clear from the survey that current exact methods cannot solve network design problems involving more than a dozen ports, and we do not expect any major break-through on exact methods in the near future. Hence, large-scale liner shipping network design will have to rely on heuristics. The current approach of splitting heuristics into a service design step, and a flow step seems promising and more research should be put into exploiting such algorithms. Whether it should be route-first-flow-next or flow-first-route-next algorithms may depend on how well the two problems interact.

Since liner shipping network design problems are highly complex, due to the large number of real-life constraints, it can be difficult to achieve useful theoretical results. It is therefore a good idea to study simplified problems that are easier to analyse. Once we understand how to solve these problems, we can add more real-life constraints. It is also the case, that many of the real-life constraints can be considered without significantly changing the problem structure, by modifying the underlying graph. Some of the basic problems could be various network flow models with setup costs on edges, and/or time constraints for the demands.

The time-space models studied in e.g. Koza et al. (2019) are showing promising results for small-/medium-scale problems, and should be investigated further in future research. In particular, the models allow various time-dependant constraints to be handled.

Finally, in the airline industry, we see a tendency that carriers operating complex networks are slowly being replaced by carriers operating only point-to-point flights. It could be interesting to study network design where services are restricted to simple routes involving only a few ports. Having fewer intermediate port calls on a service also means that the vessels can sail at lower speed, and hence reduce fuel costs and emission.

References

- Germanische Lloyd. <http://www.balkanlloyd.com/news/96-germanischer-lloyd-has-conducted-research-showing-that-new-and-efficient-4-500-teu-panamax>, 2017. [Online; accessed February 20, 2017].
- R. Agarwal and Ö. Ergun. Ship scheduling and network design for cargo routing in liner shipping. *Transportation Science*, 42(2):175–196, 2008.
- R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.
- J. F. Álvarez. Joint routing and deployment of a fleet of container vessels. *Maritime Economics & Logistics*, 11(2):186–208, 2009.
- H. Andersson, K. Fagerholt, and K. Hobbessland. Integrated maritime fleet deployment and speed optimization: Case study from ro-ro shipping. *Computers & Operations Research*, 55:233–240, 2015.
- C. Archetti and M. G. Speranza. A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, 2(4):223–246, 2014.
- A. Balakrishnan and C. V. Karsten. Container shipping service selection and cargo routing with transshipment limits. *European Journal of Operational Research*, 263(2):652–663, 2017.
- B. D. Brouer, J. Dirksen, D. Pisinger, C. E. M. Plum, and B. Vaaben. The vessel schedule recovery problem (VSRP) a MIP model for handling disruptions in liner shipping. *European Journal of Operational Research*, 224:362–374, 2013.
- B. D. Brouer, J. F. Álvarez, 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, G. Desaulniers, C.V. Karsten, and D. Pisinger. A matheuristic for the liner shipping network design problem with transit time restrictions. In F. Corman, S. Voß, and R. R. Negenborn, editors, *Computational Logistics*, pages 195–208. Springer, 2015.
- B. D. Brouer, C.V. Karsten, and D. Pisinger. Big data optimization in maritime logistics. In A Emrouznejad, editor, *Big Data Optimization: Recent Developments and Challenges*, volume 18 of *Studies in Big Data*, pages 319–344. Springer International Publishing, 2016.
- B. D. Brouer, C. V. Karsten, and D. Pisinger. Optimization in liner shipping. *4OR*, 15(1):1–35, 2017.
- S. Chandra, K. Fagerholt, and M. Christiansen. Maritime fleet deployment in ro-ro shipping under inventory constraints. *Procedia-Social and Behavioral Sciences*, 189:362–375, 2015.
- S. Chandra, M. Christiansen, and K. Fagerholt. Combined fleet deployment and inventory management in roll-on/roll-off shipping. *Transportation Research Part E: Logistics and Transportation Review*, 92:43–55, 2016.
- 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, 2013.
- M. Christiansen, E. Hellsten, D. Pisinger, D. Sacramento, and C. Vilhelmsen. Liner shipping network design. In T. Crainic, B. Gendron, and M. Gendreau, editors, *Network Design*. 2019.
- B. Dong, T. Bektaş, S. Chandra, M. Christiansen, and K. Fagerholt. A new formulation for the combined maritime fleet deployment and inventory management problem. In *International Conference on Computational Logistics*, pages 321–335. Springer, 2017.
- K. Fagerholt, T. A. V. Johnsen, and H. Lindstad. Fleet deployment in liner shipping: a case study. *Maritime Policy & Management*, 36(5):397–409, 2009.
- M. Fischetti, J. J. Salazar González, and P. Toth. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. *Operations Research*, 45(3):378–394, 1997.
- S. Gelareh, S. Nickel, and D. Pisinger. Liner shipping hub network design in a competitive environment. *Transportation Research Part E: Logistics and Transportation Review*, 46(6):991–1004, 2010.
- S. Guericke and K. Tierney. Liner shipping cargo allocation with service levels and speed optimization. *Transportation Research Part E: Logistics and Transportation Review*, 84:40–60, 2015.
- P. Hansen and N. Mladenovic. Variable neighborhood search. In E. K. Burke and G. Kendall, editors, *Search Methodologies - Introductory Tutorials in Optimization and Decision Support Techniques*, pages 313–337. Springer, 2014.
- E. Hellsten, D. Pisinger, D. Sacramento, and C. Vilhelmsen. Green liner shipping network design. In H. Psaraftis, editor, *Sustainable shipping: a cross disciplinary view*, page to appear. Springer, 2018.
- ISL. Shipping statistics and market review. Technical report, Institute of Shipping Economics and Logistics, 2016.
- C. V. Karsten. Competitive liner shipping network design. *Ph. D. Thesis, DTU Management Engineering*, 2015.
- C. V. Karsten, D. Pisinger, S. Røpke, and B. D. Brouer. The time constrained multi-commodity network flow problem and its application to liner shipping network design. *Transportation Research Part E: Logistics and Transportation Review*, 76:122–138, 2015.
- C. V. Karsten, B. D. Brouer, and D. Pisinger. Competitive liner shipping network design. *Computers & Operations Research*, 87:125 – 136, 2017a.
- C. V. Karsten, B.D. Brouer, G. Desaulniers, and D. Pisinger. Time constrained liner shipping network design. *Transportation Research. Part E: Logistics and Transportation Review*, 105:152–162, 2017b.
- K. H. Kjeldsen. Classification of ship routing and scheduling problems in liner shipping. *INFOR: Information Systems and Operational Research*, 49(2):139–152, 2011.
- D.F. Koza, G. Desaulniers, and S. Røpke. Integrated liner shipping network design and scheduling. *Transportation Science*, Accepted:1–22, 2019.

- A. Krogsgaard, D. Pisinger, and J. Thorsen. A flow-first route-next heuristic for liner shipping network design. *Networks*, 2018. In press.
- C-Y. Lee and D-P. Song. Ocean container transport in global supply chains: Overview and research opportunities. *Transportation Research Part B: Methodological*, 95:442–474, 2017.
- Q. Meng and S. Wang. Optimal operating strategy for a long-haul liner service route. *European Journal of Operational Research*, 215(1):105–114, 2011a.
- 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, 9 2011b.
- Q. Meng, S. Wang, H. Andersson, and K. Thun. Containership routing and scheduling in liner shipping: overview and future research directions. *Transportation Science*, 48:265–280, 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.
- B. Montreuil. Towards a physical internet: meeting the global logistics sustainability grand challenge. *Logistics Research*, 3:71–87, 2011.
- J. Mulder and R. Dekker. Methods for strategic liner shipping network design. *European Journal of Operational Research*, 235(2):367–377, 2014.
- R. Neamatian Monemi and S. Gelareh. Network design, fleet deployment and empty repositioning in liner shipping. *Transportation Research Part E*, 108:60–79, 2017.
- T. E. Notteboom and B. Vernimmen. The effect of high fuel costs on liner service configuration in container shipping. *Journal of Transport Geography*, 17(5):325–337, 2009.
- G. Pantuso, K. Fagerholt, and S. W. Wallace. Uncertainty in fleet renewal: a case from maritime transportation. *Transportation Science*, 50(2):390–407, 2015.
- C. E. M. Plum, D. Pisinger, J. J. Salazar-González, and M. M. Sigurd. Single liner shipping service design. *Computers & Operations Research*, 45:1–6, 2014a.
- 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, 2014b.
- H. N. Psaraftis and C. A. Kontovas. Slow steaming in maritime transportation: Fundamentals, trade-offs, and decision models. In C-Y. Lee and Q. Meng, editors, *Handbook of Ocean Container Transport Logistics: Making Global Supply Chains Effective*, pages 315–358. Springer International Publishing, 2015.
- L. B. Reinhardt and D. Pisinger. A branch and cut algorithm for the container shipping network design problem. *Flexible Services and Manufacturing Journal*, 24(3):349–374, 2012.
- D. Ronen. Cargo ships routing and scheduling: Survey of models and problems. *European Journal of Operational Research*, 12(2):119–126, 1983.
- D. Ronen. Ship scheduling: The last decade. *European Journal of Operational Research*, 71(3):325–333, 1993.
- Z. Sun and J. Zheng. Finding potential hub locations for liner shipping. *Transportation Research Part B: Methodological*, 93:750–761, 2016.
- K. Thun, H. Andersson, and M. Christiansen. Analyzing complex service structures in liner shipping network design. *Flexible Services and Manufacturing Journal*, 29(3-4):535–552, 2017.
- P. Toth and D. Vigo. *Vehicle routing: problems, methods and applications*, volume 18. SIAM, 2015.
- N.K. Tran and H-D. Haasis. Literature survey of network optimization in container liner shipping. *Flexible Services and Manufacturing Journal*, 27:139–179, 2015.
- Unctad. Review of maritime transport. Technical report, United Nations Conference on Trade and Development, 2017a.
- Unctad. UnctadSTAT. <http://unctadstat.unctad.org/wds/TableView/tableView.aspx>, 2017b. [Online; accessed February 9, 2017].
- S. Wang and Q. Meng. Robust schedule design for liner shipping services. *Transportation Research Part E: Logistics and Transportation Review*, 48(6):1093–1106, 2012.

- S. Wang and Q. Meng. Liner shipping network design with deadlines. *Computers & Operations Research*, 41:140–149, 2014.
- J. Zheng, Q. Meng, and Z. Sun. Impact analysis of maritime cabotage legislations on liner hub-and-spoke shipping network design. *European journal of operational research*, 234(3):874–884, 2014.

A Appendix: Notation used in this chapter

We have tried to use an uniform mathematical notation throughout the chapter, although each section needs some additional symbols. The notation has been gathered and presented in the following tables. First, the general notation presented in the introduction that will be commonly used in the mathematical models of the chapter will be compiled. Next, the extra notation needed to define the models of each specific section is presented.

Table 5. General Notation from the Introduction

<u>Sets</u>
N := Set of Ports
A := Set of Sailing Arcs
K := Set of Commodities
V := Set of Vessel Classes
S := Set of Services
<u>Parameters</u>
o_k := Origin port for commodity $k \in K$
d_k := Destination port for commodity $k \in K$
q_k := Quantity of commodity $k \in K$
u_v := Cargo capacity for vessel class $v \in V$
m_v := Available fleet quantity of vessel class $v \in V$
$\xi_i^k := \begin{cases} q_k & \text{if port } i \in N \text{ is the origin port of demand } k \in K \\ -q_k & \text{if port } i \in N \text{ is the destination port of demand } k \in K \\ 0 & \text{otherwise.} \end{cases}$
c_{ij}^k := Unit-cost for transporting a unit of commodity $k \in K$ through arc $(i, j) \in A$
c^v := Cost for deployment of a vessel from vessel class $v \in V$
c_{ij}^v := Sailing cost for vessel class $v \in V$ traversing arc $(i, j) \in A$
c_{ik}^T := Cost per unit of commodity $k \in K$ transshipped in port $i \in N$
t_{ij}^v := Sailing time for vessel class $v \in V$ traversing arc $(i, j) \in A$
b_i := Berthing time for the port call $i \in N$
c_s := Cost for operation service $s \in S$.
m_v^s := Required number of vessels from vessel class $v \in V$ to maintain the weekly frequency in service $s \in S$

Table 6. Additional notation for Section 3.1

<u>Parameters</u>
u_{ij}^s := Capacity for service $s \in S$ along arc $(i, j) \in A$
<u>Decision variables</u>
$x_{ij}^{ks} \in \mathbb{R}^+ :=$ Amount of commodity $k \in K$ transported in service $s \in S$ along arc $(i, j) \in A$
$y_s \in \{0, 1\} :=$ Equal to 1 if service $s \in S$ is selected in the network, and 0 otherwise

Table 7. Additional notation for Section 3.2

<u>Sets</u>	
S^v	:= Set of maximum number of services for vessel class $v \in V$
<u>Decision variables</u>	
$x_{ij}^{ks} \in \mathbb{R}^+$:= Amount of commodity $k \in K$ transported in service $s \in S^v$ along arc $(i, j) \in A$
$\tau_i^s \in \mathbb{R}^+$:= Departure time from port $i \in N$ of the vessel operating service $s \in S^v$
$w^s \in \mathbb{Z}^+$:= Number of deployed vessels to maintain a weekly frequency in service $s \in S^v$
$y_{ij}^s \in \{0, 1\}$:= Equal to 1 if arc $(i, j) \in A$ is selected on the service $s \in S^v$, and 0 otherwise

Table 8. Additional notation for Section 3.2 (Reinhardt and Pisinger (2012))

<u>Parameters</u>	
t_{max}	:= Length of the time horizon
<u>Decision variables</u>	
$x_{ij}^{kv} \in \mathbb{R}^+$:= Amount of commodity $k \in K$ transported by vessel $v \in V$ along arc $(i, j) \in A$
$f_j^{kv} \in \mathbb{R}^+$:= Amount of commodity $k \in K$ transshipped at port $j \in N$ by vessel $v \in V$
$f_{jih}^{kv} \in \mathbb{R}^+$:= Amount of commodity $k \in K$ arriving at port $i \in N$ through arc $(j, i) \in A$ and not leaving in arc $(i, h) \in A$ by vessel $v \in V$
$e_{ij}^v \in \mathbb{Z}^+$:= Position of arc $(i, j) \in A$ in the service of vessel $v \in V$
$\tau_v \in \mathbb{R}^+$:= Route length of the service operated by vessel $v \in V$
$y_{ij}^v \in \{0, 1\}$:= Equal to 1 if arc $(i, j) \in A$ is selected in the service for vessel $v \in V$, and 0 otherwise
$z_{ij}^v \in \{0, 1\}$:= Equal to 1 if arc $(i, j) \in A$ is either the first or the last arc in the service for vessel $v \in V$, and 0 otherwise
$\gamma_i^v \in \{0, 1\}$:= Equal to 1 if port $i \in N$ is the hub-port in the service for vessel $v \in V$, and 0 otherwise
$\lambda^v \in \{0, 1\}$:= Equal to 1 if vessel $v \in V$ is deployed for operating a service, and 0 otherwise

Table 9. Additional notation for Section 4.1

<u>Sets</u>	
P^k	:= Set of paths for commodity $k \in K$
P_a	:= Set of paths using arc $a \in A$
<u>Parameters</u>	
u_{ij}/u_a	:= Flow capacity through arch $a = (i, j) \in A$
c_p^k	:= Unit flow cost of commodity $k \in K$ through path $p \in P_k$
<u>Decision variables</u>	
$x_{ij}^k \in \mathbb{R}^+$:= Flow of commodity $k \in K$ through arc $(i, j) \in A$
$f_p^k \in \mathbb{R}^+$:= Flow of commodity $k \in K$ through path $p \in P_k$

Table 10. Additional notation for Section 4

<u>Sets</u>	
N_s	:= Set of ports in service $s \in S$
\bar{N}_s	:= Set of ports available for inclusion into service $s \in S$
L_i	:= Lockset containing the ports which are forbidden to remove, if port $i \in N_s \cup \bar{N}_s$ is included (removed)
<u>Parameters</u>	
Δ_{is}^{R+}	:= Estimated revenue change from including port $i \in \bar{N}_s$ into service $s \in S$
Δ_{is}^{R-}	:= Estimated revenue change from removing port $i \in N_s$ from service $s \in S$
Δ_{is}^{T+}	:= Estimated duration change from including port $i \in \bar{N}_s$ into service $s \in S$
Δ_{is}^{T-}	:= Estimated duration change from removing port $i \in N_s$ from service $s \in S$
\bar{m}_v	:= Number of free vessels of vessel class $v \in V$
τ_s	:= Round trip time for service $s \in S$
η_s^+	:= Maximum number of inclusions into service $s \in S$
η_s^-	:= Maximum number of removals into service $s \in S$
<u>Decision variables</u>	
$x_{is}^+ \in \{0, 1\}$:= Equal to 1 if port $i \in \bar{N}_s$ is included into service $s \in S$, and 0 otherwise
$x_{is}^- \in \{0, 1\}$:= Equal to 1 if port $i \in N_s$ is removed from service $s \in S$, and 0 otherwise
$\zeta_s \in \mathbb{Z}$:= Change in number of vessels in service $s \in S$

Table 11. Additional notation for Section 5

<u>Sets</u>	
A_s	:= Set of sailing arcs for service $s \in S$
H_s	:= Full set of sub-paths for service $s \in S$ (a sub-path is a part of a route in which the container travels on a single service and is denoted $\langle i, j, s \rangle$)
A_{ij}^s	:= Set of sailing arcs of service $s \in S$ included in sub-path $\langle i, j, s \rangle$
<u>Parameters</u>	
u_a	:= Capacity for arc $a \in A_s$
c_k^R	:= Penalty cost per container for rejected demand of commodity $k \in K$
c_{ijs}^k	:= Cost of routing one container of commodity $k \in K$ on sub-path $\langle i, j, s \rangle$
h_k	:= Maximum allowed number of sub-paths on which commodity $k \in K$ can travel
<u>Decision variables</u>	
$y_s \in \{0, 1\}$:= Equal to 1 if service $s \in S$ is selected, and 0 otherwise
$x_{ijs}^{hk} \in \mathbb{R}^+$:= Flow of commodity $k \in K$ using sub-path $\langle i, j, s \rangle$ as the h^{th} stage for $s \in S$, $\langle i, j, s \rangle \in A_s$, and $h = 1, 2, \dots, h_k$
$z_k \in \mathbb{R}^+$:= Unmet demand (number of containers) for commodity $k \in K$

Table 12. Additional notation for Section 6

<u>Parameters</u>	
$c(x)$:= Concave cost function of using an edge $(i, j) \in A$ for the flow x
<u>Decision variables</u>	
$x_{ij}^k \in \mathbb{R}^+$:= Flow of commodity $k \in K$ on edge $(i, j) \in A$
