Chapter 9
Green Liner Shipping Network Design

Erik Hellsten, David Pisinger, David Sacramento, and Charlotte Vilhelmsen

Abstract *Green Liner Shipping Network Design* refers to the problems in *green logistics* related to the design of maritime services in liner shipping with a focus on reducing the environmental impact. This chapter discusses how to more efficiently plan the vessel services with the use of mathematical optimization models. A brief introduction to the main characteristics of Liner Shipping Network Design is given, as well as the different variants and assumptions that can be considered when defining this problem. The chapter also includes an overview of the algorithms and approaches that have been presented in the literature to design such networks.

Acronyms and Abbreviations

ECA	Emission Control Areas
IMO	International Maritime Organization
LNG	Liquid Natural Gas
LP	Linear Programming
LSNDP	Liner Shipping Network Design Problem
LSP	Liner Service Planning
MARPOL	International Convention for the Prevention of Pollution from Ships
MCFP	Multi-Commodity Flow Problem
MIP	Mixed Integer Programming
SSSCRP	Simultaneous Ship Scheduling and Cargo Routing Problem
TEU	Twenty-Foot Equivalent Units

E. Hellsten · D. Pisinger (✉) · D. Sacramento · C. Vilhelmsen
DTU Management Engineering, Technical University of Denmark, Kongens Lyngby, Denmark
e-mail: dapi@dtu.dk

H. N. Psaraftis (ed.), *Sustainable Shipping*,
https://doi.org/10.1007/978-3-030-04330-8_9

TSP	Traveling Salesman Problem
UNCTAD	United Nations Conference on Trade and Development
VNS	Variable Neighborhood Search
VRP	Vehicle Routing Problem

1 Introduction

The liner shipping industry is a vital part of the global economy, constituting one of the greenest modes of cargo transport. In full load, the new mega-vessels emit only 3 g of CO_2 for transporting 1 metric tonne of cargo 1 kilometer (Maersk 2017); in comparison, trains average on 18 g and flights on 560 g (see Fig. 9.1). Today, around 90% of the global trade, by volume, is carried out by seaborne transportation, a number which is expected to continue rising. During the last three decades, the volume of containerized cargo has grown by more than 8% per year, and more than 5.150 container vessels were in operation worldwide in 2017. The largest vessels carry more than 20.000 20-foot equivalent units (TEU), and during 2016, a container volume of around 140.000.000 TEU was estimated to pass through this vast network (Unctad 2017a,b). In this chapter we will show how optimization techniques can be used to design more efficient liner shipping networks in order to further decrease the environmental footprint of liner shipping.

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 Fig. 9.2), 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 ships pick up and deliver containers. From the hubs, the containers are then transported to other ports by smaller, more flexible vessels, called *feeder vessels*. *Transshipments* occur both between larger vessels and smaller vessels but also between larger vessels when no suitable service connects the origin and destination

Fig. 9.1 Estimated CO_2 emission for transporting 1 tonne of goods 1 kilometer for different transportation modes (Source: Maersk 2017)

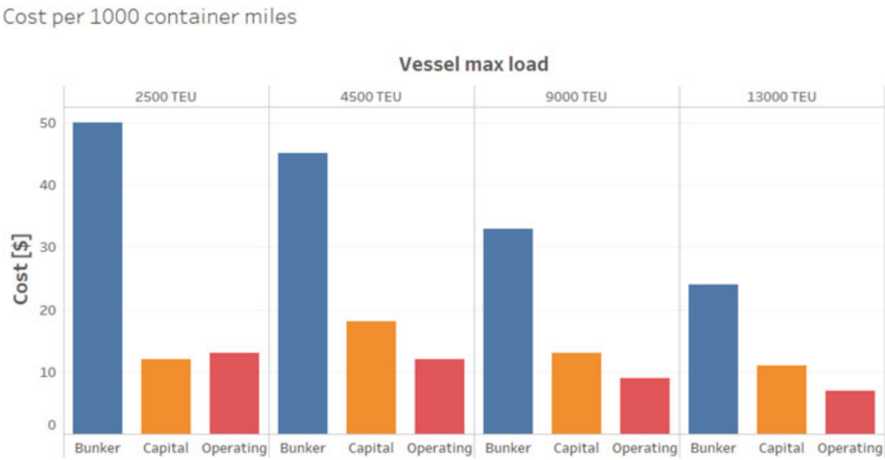

Fig. 9.2 Estimated cost per 1000 container miles for different vessel sizes. The vessels are assumed to sail at 19 knots and the bunker price is estimated as 750$/tonne. We see that bunker represents the largest cost and that transporting containers on larger vessels requires significantly less fuel (Source: Germanischer Lloyd)

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.

Another major constraint in liner shipping is *cabotage rules*. To protect the national trade business, many countries forbid foreign carriers to ship cargo between two ports within the country. See Brouer et al. (2014a) for examples of cabotage rules.

The major costs for the carriers are vessel acquirement and bunker. But other costs, like canal fees, port costs and transshipment costs, are also highly significant. Most papers in the literature presumes that fuel consumption is frequently estimated as a cubic function of the speed (see Fig. 9.3). Psaraftis and Kontovas (2013) point out that the fuel consumption is given as a complex function depending on many ship parameters and that the cubic approximation on terms of the sailing speed is valid for tankers and bulk carriers, whereas higher exponents should be considered in liner shipping. As the speed has such an impact on the fuel consumption, *slow steaming* is often used to reduce the consumption, i.e., operating the container vessels at speeds, significantly lower than their maximum speed. Especially after the financial crisis in 2008, maritime shipping companies implemented slow steaming policies for cost-cutting purposes. The drawback with slow steaming is, however, that more vessels are required to keep the regularity with respect to weekly departures and also that transit times become longer, yielding a lower level of service for the customers. In general, services has two directions, *head-* and *backhaul*, where most of the cargo is transported in the head-haul direction. A good example of this is the trade between Europe and Asia, where most of the goods are delivered

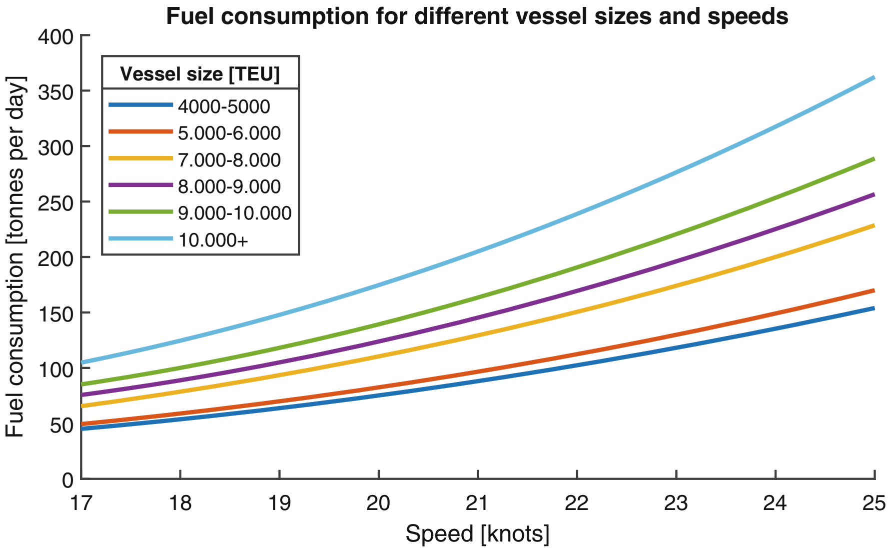

Fig. 9.3 Estimated fuel consumption as a function of steaming speed and vessel size (Source: Notteboom and Vernimmen 2009)

from Asia to Europe. In this case, vessels are slow steaming in the backhaul direction where less customers are affected by the increased transit time.

Due to the ability to transport large numbers of containers with each vessel, liner shipping is one of the most energy-efficient transportation forms. Nonetheless, due to the large volumes transported, the shipping industry contributes significantly to the global CO_2 emissions. According to the *IMO Green House Gas Study 2014*, in 2012 the international shipping industry was estimated to account for 2.2% of the global greenhouse gas emissions, of which approximately a quarter was caused by container vessels, which corresponds to around a billion tonnes of CO_2 annually. These emissions are further expected to increase between 50% and 250% in the next 30 years (IMO 2014).

Although liner shipping is the most efficient transportation mode in terms of CO_2, the vessels commonly operate using "dirty" fuel, emitting various pollutants which are harmful for the environment and the human health. In 2013, it was estimated that, in Europe, ships accounted for 18% of the nitrogen oxides (NO_x), 18% of the sulfur oxides (SO_x) and 11% of the particle matter ($PM_{2.5}$) of the total annual emissions, respectively (Wan et al. 2016). Measures to control SO_x and PM emissions are being applied through the International Convention for the Prevention of Pollution from Ships (also known as MARPOL) and Emission Control Areas (ECA). The emission percentages, of these gases, from seaborne trade are currently much higher than from other modes of transportation, such as rail or aviation. The maximum permitted level of sulfur content contained in marine fuels is currently 3.5%, but it will be reduced to 0.5% by 2020 (IMO 2014).

There are several measures which could be applied to counteract these polluting emissions in the maritime industry. Cleaner practices and maritime policies should be imposed, both by industry and by governments, to control the environmental impact. It is important, however, to emphasize that maritime companies follow long-term strategic plans, where the vessel fleet has a long life expectancy, around 25 to 30 years, since building new vessels is a huge investment. Therefore, it takes a long time before green innovations regarding engines or vessel design can be applied in practice.

Instead, one of the major roads toward a greener shipping industry must be through more efficient utilization of the current assets. If a more efficient service structure can be developed, the same vessels could transport the same amount of cargo while running at a lower speed. This far, the literature on pure green liner shipping network design is highly limited. However, as the bunker cost is one of the major costs for the carriers, reducing the cost is strongly correlated to reducing the fuel consumption. Hence, reducing the cost can, indirectly, be seen as contributing to the green objective. Further, increasing the level of service would likely result in that transportation changes to shipping from other modes. As the CO_2 emissions of liner shipping are lower, an increase in the level of service could also be expected to result in a more sustainable overall transportation system. All in all, to make a greener shipping industry, it would be of great value to develop models, solution algorithms and decision support tools for liner shipping network design.

1.1 Liner Shipping Network Design Problem

The Liner Shipping Network Design Problem (LSNDP) can informally be defined as follows: given a collection of ports, a fleet of container vessels and a group of origin-destination demands, construct a set of services for the container vessels such that the overall operational expenses are minimized while ensuring that all demands can be routed through the resulting network, respecting the capacity of the vessels.

In this section, we present some notation of the LSNDP. For a complete model, 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 two ports. To each port, there is a corresponding port call cost c^i_{P}, as well as a berthing time b_i, for the port call. The set of commodities to transport is denoted by K, and for each commodity $k \in K$, there is an origin port o_k, a destination port d_k and a quantity δ_k measured in TEUs. Finally, the set V denotes the set of *vessel classes* with the corresponding cargo capacities q_v, available quantity M_v, fuel consumption g_v per nautical mile and additional speed limitations. Furthermore, for convenience, the demand of the commodities in the ports are defined as:

$$\xi_i^k = \begin{cases} \delta_k & \text{if port } i \text{ is the origin port of commodity } k \\ -\delta_k & \text{if port } i \text{ is the destination port of commodity } k \\ 0 & \text{otherwise.} \end{cases} \tag{9.1}$$

There is a limited fleet of container vessels, but not all vessels need to be used. The deployment of a vessel has an associated *charter cost* c^v. Additionally, there exist other costs related to the resulting network, such as the *sailing cost* c^v_{ij} associated with each vessel and each arc, which is given as a combination of the port call cost c^j_{P} and the fuel consumption for the corresponding leg. Furthermore, handling costs of containers in the ports are considered as well, incurring a cost c^i_{L} (c^i_{U}) per unit of container (un)loaded in the port. Containers can also be transferred from one vessel to another in the ports, which incurs a unit *transshipment cost* $c^{i,k}_{\mathrm{T}}$.

One of the main traits of the liner shipping industry is the regular operation of services under a pre-established schedule. It is imposed that all services should have *weekly operations*, meaning that if a round trip takes 8 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. In addition, services must be cyclic, visiting a sequence of ports before returning to the original port. However, a service is allowed to be non-simple, meaning that a port can be visited several times, since this may improve transit times. Services where only one port is visited twice are called *butterfly services*, and the port which is visited twice is denoted the *butterfly port*.

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

- *Transit time constraints* As described above, the transit time of each commodity has an associated time limit that must be respected. If the transit time is not respected, perishable goods may become unsalable. Many early models for LSNDP did not consider this constraint.
- *Transshipment costs* Several early models for LSNDP did not consider transshipment costs. However, 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 formulation of LSNDP states that all commodities must be flowed through the network, many models allow rejection of commodities and instead impose a penalty.
- *Speed optimization* There are three main categories of models regarding speed optimization: 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.

Most models for the LSNDP design a network without a specific schedule. Hence the route 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 optimization problems, see the survey papers Ronen (1983, 1993), Christiansen et al. (2004, 2013), Meng et al. (2014), Brouer et al. (2016, 2017) and Lee and Song (2017).

1.2 Measuring and Calculating Transportation Emissions

The environmental effects associated with the maritime industry are becoming a major concern. The large amount of pollution produced by container vessels has not gone unnoticed, due to considerable emissions of various types of pollutants such as SO_x, NO_x, PM and CO_2. The International Maritime Organization (IMO) is investigating the possibility of reducing these emissions by establishing regulatory policies.

The maritime industry is an economy-dependent industry, and the minimization of the operational cost is paramount. As noted by Notteboom (2006), the price of fossil fuels is one of the largest in maritime transportation. Ronen (2011) estimates that the bunker cost makes up more than 75% of the total operating cost of a vessel. The fuel cost is strongly related to the operating speed of the vessels, where there exists an important trade-off. Based on this, the estimation of greenhouse gases such as CO_2 can be given by an energy approach, which can be obtained from the fuel consumption and an appropriate emission factor to convert carbon content of the fuel into CO_2 emissions. These conversion factors have been established by IMO according to the type of fuel used by the container vessel (IMO 2014). The default values are given on the basis of gram CO_2 per gram fuel, being 3.114 g CO_2/g for heavy fuel oil and 3.206 g CO_2/g for marine diesel and marine gas oils. An estimate E_{ijv} of the total CO_2 emissions for a vessel v in a leg trip between port i and j can be obtained as:

$$E_{ijv} = \sum_{z \in Z} \alpha_{v,z} \left[g_{\mathrm{S}}^{v,z} \left(\frac{s_{ij}^{v}}{s_{v}^{*}} \right)^{n} d_{ij}^{z} + g_{\mathrm{I}}^{v,z} b_{j}^{z} \right] \tag{9.2}$$

where Z is the set of bunker types, indexed by z; $\alpha_{v,z}$ is the corresponding conversion factor for vessel v according to the type of fuel z; $g_{\mathrm{S}}^{v,z}$ and $g_{\mathrm{I}}^{v,z}$ is the fuel consumption of vessel v when sailing and idle at the port with bunker type z, respectively; s_{ij}^{v} is the operational speed of the vessel between the ports; and s_{v}^{*} is the design speed of the vessel. The exponent n is usually approximated to be around 3, meaning that the fuel consumption varies cubically with the speed (Stopford 2009). Moreover, d_{ij}^{z} is the sailing distance between the ports in nautical miles, and b_{j}^{z} is the berthing time at the port j for the vessel with bunker type z. This estimate is a simple representation of how CO_2 emissions can be calculated for its incorporation into a mathematical model.

Although sustainable maritime transportation is gaining more importance in Operations Research, the literature is still very scarce. In the context of routing and scheduling, there are a few papers dealing with green maritime transportation. Kontovas (2014) presents different approaches that can be considered when incorporating environmental dimensions: through the minimization of total emissions, internalizing the external cost of emissions and adding constraints to limit the produced emissions. The author remarks that minimizing fuel consumption is not equivalent to minimizing the total emissions, since vessels are generally equipped with main and auxiliary engines, which usually use different types of fuel. Another way to reduce the greenhouse gas emissions is to introduce ECAs, which are predefined areas where vessels are not allowed to use fuels with high sulfur content. Fagerholt et al. (2015) and Dithmer et al. (2017) present mathematical formulations introducing these emission control regulations. In the latter case, in a similar way as described in Kontovas (2014), the authors also study the approach of internalizing the external costs of emissions, making it possible to analyze the routing and scheduling of the services if a tax system is implemented in the future.

1.3 The LINER-LIB Test Instances

In order to make it easier to compare algorithms for liner shipping network, (Brouer et al. 2014a) published 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. They range from smaller networks suitable for optimal methods to large-scale instances spanning the globe. Table 9.1 gives an overview of these instances.

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. For the high capacity case, the adjustments are reversed.

Currently, most papers only report results for the six first instances, with (Krogsgaard et al. 2018) being the only to report results for the *WorldLarge* instance.

Table 9.1 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)

Instance	Category	$\|N\|$	$\|K\|$	$\|V\|$	min v	max v
Baltic	Single-hub	12	22	2	5	7
West Africa (WAF)	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	4000	6	140	212
WorldSmall	Multi-hub	47	1764	6	209	317
WorldLarge	Multi-hub	197	9630	6	401	601

1.4 Outline

This chapter is organized as follows. In Sect. 2 we discuss the challenges in designing an energy-efficient liner shipping network and show that algorithms roughly can be split into four different families, which are studied in Sects. 3, 4, 5, and 6. In Sect. 3 we give an overview of integrated MIP models, while Sect. 4 studies two-stage algorithms where the routes 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 routes. In Sect. 6 we consider algorithms based on first flowing containers and then designing routes. Finally, Sect. 7 shows how speed optimization can be used to lower energy consumption in liner shipping. The chapter is concluded in Sect. 8 with a short discussion of future trends and challenges.

2 Overview of Algorithms

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

The problem is further complicated by the fact that ports often are visited several times in the same route. This is obviously the case for pendulum routes where a vessel is sailing back and forth along the same route, but multiple visits to a port (typically a hub) often take place to ensure faster transportation times. However, formulating the problem as 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 are transshipped and at which costs. This adds further complexity to the problem and makes a graph 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 routes and flows containers through the resulting network. In order to handle this task, two sets of variables are needed: variables to select edges in a route and variables to denote the flow on each edge. 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. (2014) and Wang and Meng (2014).
- *Two-stage algorithms* As the name suggests, these algorithms solve the problem in two steps: designing the routes and flowing containers through the resulting network. Frequently, these algorithms contain a feedback mechanism, where output from the second-stage flow model is used as input to improve the routes in the first stage. Successful applications of this approach include Agarwal and Ergun (2008), Álvarez (2009), Brouer et al. (2014a,b), 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 method for generating a network by having a list of candidate routes as input. The idea behind these algorithms is to use the experience from existing planners to design a large number of promising candidate routes. The algorithm then selects a subset of the candidate routes 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 route may be a useful method.
- *Backbone flow* The idea behind this approach is that it can be difficult to design the individual routes without knowing how the containers will flow through the network. Hence 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 fewer 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 multisegment 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 smaller instances can be solved to proven optimality within a reasonable time frame; hence the algorithms will often return a suboptimal solution. The subset-of-routes-based algorithms also solve the problem to optimality given that only the proposed candidate routes are valid. In practice, however, there may be an exponential number of valid routes, and we cannot expect to get all routes as input. If only a subset of all valid routes is given as input, the found solution may be suboptimal. The two-stage algorithms and backbone-network algorithms are both heuristics, since they first solve one stage and then optimize the second stage with the first-stage decisions fixed.

3 Mixed Integer Programming Models

The design of a liner shipping network includes numerous decisions, such as the routing of containers, the fleet deployment and the service design. The design of shipping networks is beyond the limited capacity of human planners, and it requires the use of several complex decision support tools. Mixed Integer Programming and graph-based models will be used in the subsequent sections to define the network design problem mathematically. Several MIP formulations of the LSNDP have been proposed during the last decades. We will give an overview of some of the formulations and discuss their advantages and limitations.

3.1 Service Formulation for LSNDP

Liner Shipping is based on the operation of services, which are defined by a sequence of ports that are visited by the vessel under a previously established schedule. The main objective of LSNDP is to design the shipping network by selecting services for the vessels so that the demand can be flowed at minimum cost while the overall benefit is maximized. Considering this fact, the first mathematical formulation is introduced in this section, which models the problem based on a service formulation, i.e. where the set of all feasible services are predefined in the model.

Before introducing the mathematical models presented in the literature, we briefly introduce a simple mathematical model based on a service formulation for better understanding. We will consider the notation presented in the introduction in Sect. 1.1 but with a small extension. Let $G = (N, A)$ be a directed graph, where N is the set of ports and A is the set of arcs connecting the ports. We now define the set S as the set of all feasible services in the model. Notice that S may be exponentially large. Let c_s be the cost of operating service $s \in S$, c_{ij}^{k} the unit cost per commodity $k \in K$ for traversing arc $(i, j) \in A$, $v(s)$ the corresponding vessel class $v \in V$ for the service $s \in S$, $m_{v(s)}$ the required number of vessels of class $v(s)$ for operating service $s \in S$ and a_{ij}^{s} a binary parameter indicating if the arc $(i, j) \in A$ is traversed in service $s \in S$. Finally, let x_{ij}^{ks} be a continuous variable indicating the amount of commodity $k \in K$ transported in service $s \in S$ through the arc $(i, j) \in A$ and y_s a binary variable for the selection of service $s \in S$ in the network. Now, the service formulation of the LSNDP can be expressed as:

$$\min \quad \sum_{s \in S} c_s y_s + \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k \sum_{s \in S} x_{ij}^{ks} \tag{9.3a}$$

$$s.t. \quad \sum_{s \in S} \sum_{j:(i,j) \in A} x_{ij}^{ks} - \sum_{s \in S} \sum_{j:(j,i) \in A} x_{ji}^{ks} = \xi_i^k \qquad i \in N, k \in K \tag{9.3b}$$

$$\sum_{\substack{s \in S \\ v(s)=v}} m_{v(s)} y_s \le M_v \qquad v \in V \tag{9.3c}$$

$$\sum_{k \in K} x_{ij}^{ks} \le q_{v(s)} a_{ij}^s y_s \qquad (i, j) \in A, s \in S \tag{9.3d}$$

$$x_{ij}^{ks} \ge 0 \qquad (i, j) \in A, k \in K, s \in S \tag{9.3e}$$

$$y_s \in \{0, 1\} \qquad s \in S. \tag{9.3f}$$

The objective function (9.3a) minimizes the total operational cost of the network. The first term accounts for the total fixed cost of the selected services, whereas the second term accounts for the sailing cost of shipping the demand. Constraints (9.3b) are the flow conservation constraints, constraints (9.3c) ensure that the deployed vessels on the services do not exceed the available fleet, and the flow capacity of the selected services has to be respected, which is described by constraints (9.3d). Finally, the domain of the variables is defined by constraints (9.3e) and (9.3f).

A successful implementation, based on a service formulation, was presented by Álvarez (2009). Álvarez extends the previous formulation to define the Liner Shipping Network Design at the tactical level, where the formulation combines the routing and deployment of a fleet of container vessels. The formulation relies on the set of all feasible services, which are given as a combination of a vessel type, its corresponding speed and the route structure. Therefore, it is possible to accommodate services that are proposed externally by the planners as services generated internally by a solution algorithm, meaning that any type of non-simple services can be considered in the set S of services. However, as the size of the problem increases, the number of feasible services in the problem grows exponentially, making the model intractable to solve.

Moreover, for a better utilization of the capacity of the vessels, the model allows the rejection of cargo incurring a goodwill penalty, where continuous variables are defined to account for the demand that is delivered and rejected by the liner company. With reference to the above, Álvarez also defines continuous variables for the amount of cargo that is transported along an arc on a service as well as continuous variables for different operations of loading and unloading containers in ports on specific services. These variables can be used to identify the amount of containers that are transshipped between services. 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, using integer variables to control the amount of vessels deployed for a chosen service. 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 formulations to consider transshipment when designing the shipping network.

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. This process can be inefficient and very time-consuming. Therefore, an alternative mathematical formulation is introduced in this section, which is based on an arc formulation. The set of services S is no longer considered in the problem, but the services are instead designed as part of the problem.

Next, we will present a simple mathematical model based on an arc formulation. For this we will again use the notation presented in the introduction, Sect. 1.1, with small extensions. Let $G = (N, A)$ be a directed graph, where N is the set of ports and A is the set of arcs connecting the ports. Moreover, let V be defined as the set of vessel classes and c^v the cost for deploying a vessel belonging to class $v \in V$. We will introduce the set S^v as the set of services for the vessel class $v \in V$. We also introduce t_{ij}^v and c_{ij}^v as the sailing time and cost by a vessel of type $v \in V$ traversing arc $(i, j) \in A$, respectively, and b_j the berthing time at port $j \in N$. Finally, let x_{ij}^{ks} be a continuous variable denoting the flow of a commodity $k \in K$ on an arc $(i, j) \in A$ in the service $s \in S^v$ belonging to vessel class $v \in V$, y_{ij}^{sv} a binary variable for the selection of an arc $(i, j) \in A$ in the service $s \in S^v$ operated by the vessel class $v \in V$, τ_i^s a continuous variable for the time in service $s \in S^v$ of a vessel class $v \in V$ arriving at port $i \in N$, and m_s^v an integer variable indicating the number of vessels from class $v \in V$ needed to be deployed to maintain the weekly frequency in the service $s \in S^v$. Now, the arc formulation of the LSNDP can be expressed as follows:

$$\min \sum_{v \in V} \sum_{s \in S^v} c^v m_s^v + \sum_{v \in V} \sum_{s \in S^v} \sum_{(i,j) \in A} c_{ij}^v y_{ij}^{sv} + \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k \sum_{v \in V} \sum_{s \in S^v} x_{ij}^{ks} \quad (9.4a)$$

$$s.t. \sum_{v \in V} \sum_{s \in S^v} \sum_{j:(i,j) \in A} x_{ij}^{ks} - \sum_{v \in V} \sum_{s \in S^v} \sum_{j:(j,i) \in A} x_{ji}^{ks} = \xi_i^k \qquad i \in N, k \in K \quad (9.4b)$$

$$\sum_{j:(i,j) \in A} y_{ij}^{sv} - \sum_{j:(j,i) \in A} y_{ji}^{sv} = 0 \qquad i \in N, v \in V, s \in S^v \quad (9.4c)$$

$$\sum_{k \in K} x_{ij}^{ks} \leq q_v \cdot y_{ij}^{sv} \qquad (i,j) \in A, v \in V, s \in S^v \tag{9.4d}$$

$$\tau_j^s \geq (\tau_i^s + t_{ij}^v + b_j) y_{ij}^{sv} \qquad i, j \in N, v \in V, s \in S^v \tag{9.4e}$$

$$\sum_{(i,j) \in A} y_{ij}^{sv}(t_{ij}^v + b_j) \leq 24 \cdot 7 \cdot m_s \qquad v \in V, s \in S^v \tag{9.4f}$$

$$\sum_{s \in S^v} m_s^v \leq M_v \qquad v \in V \tag{9.4g}$$

$$x_{ij}^{ks} \geq 0 \qquad (i,j) \in A, k \in K, v \in V, s \in S^v \tag{9.4h}$$

$$y_{ij}^{sv} \in \{0, 1\} \qquad (i,j) \in A, v \in V, s \in S^v \tag{9.4i}$$

$$m_s^v \in \mathbb{Z}^+ \qquad v \in V, s \in S^v \tag{9.4j}$$

$$\tau_i^s \geq 0 \qquad i \in N, v \in V, s \in S^v \tag{9.4k}$$

The objective function (9.4a) minimizes the cost of deploying the vessels and designing the services and the cost for transporting the commodities through the network. The flow conservation constraints for the cargo variables are given in constraints (9.4b), whereas the flow conservation constraints for the routing variables are given in constraints (9.4c). The flow of cargo on an edge (i, j) cannot exceed the capacity q_v of a vessel, as expressed in (9.4d). If the vessel is not used for the given edge, i.e., $y_{ij}^{sv} = 0$, then the capacity is zero. The subtour elimination constraints for the routing variables are given by the time variables in constraints (9.4e). Note that it is required to linearize these constraints, as they are non-linear. Moreover, the weekly frequency of the services and the deployment of the fleet is limited by constraints (9.4f). The availability of the fleet is limited by constraints (9.4g). Finally the domain of the variables is defined in constraints (9.4h), (9.4i), (9.4j) and (9.4k).

The model presented above is a simple representation of the arc formulation for the LSNDP. 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 consider all the assumptions that can occur in Liner Shipping. Reinhardt and Pisinger (2012) proposed a MIP model based on an arc-flow formulation where the network design and the fleet assignment are combined; however, in this case, cargo rejection is not considered. As argued in Agarwal and Ergun (2008), transshipment is the core of liner shipping; hence, these operations should not be ignored when designing the network. Reinhardt and Pisinger (2012) include these operations into the formulation and accounts correctly for the transshipment cost in the intermediate ports. Moreover, one of the main considerations of the model is the inclusion of butterfly services, where it is allowed to visit a single port twice during the service. Due to the allowance of butterfly services, the model requires the definition of extra binary variables for the identification of the unique centre point, i.e. the hub port in the vessel route, and the finding of the first and last arc visiting the hub port, respectively. Similarly, as proposed by Miller et al. (1960), positive integer variables are defined for enumerating the arcs in the vessel route to avoid the formation of subtours in the services. The definition of these variables will be used to model the transshipment of cargo in hub ports. Furthermore, the model also considers the fleet deployment with a heterogeneous fleet. It is possible to define service-dependent capacities according to the time horizon and the frequency.

The high level of detail in the model allows a fairly realistic representation of the problem, making it possible to design efficient services reducing the overall operational costs and CO_2 emissions. Nonetheless, the model is NP-hard and also in practice very difficult to solve. The model includes several "big-M" constraints, resulting in weak bounds from LP relaxation. The proposed method to solve this problem is Branch-and-Cut, as it has presented good results to the VRP and other transportation network design problems. The idea is to solve the previous relaxed problem without the transshipment constraints and the connectivity constraints in butterfly nodes and, then, gradually add cuts to the formulation whenever they are violated. The implemented method is tested against the CPLEX MIP solver on a set of test instance with up to 15 ports. The results show that the developed branch-and-cut method clearly outperforms the solver, even though some test instances are not solved to optimality. This method is not suitable for solving real-life instances such as LINER-LIB; however it provides promising results for smaller feeder services in liner shipping network design problems.

3.3 Port Call Formulation for LSNDP

The majority of the models for LSNDP are defined using an arc formulation, but such formulations 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). Alternatively, the problem can be defined with a service formulation, but the number of variables will increase exponentially with the size of the problem, as seen in Álvarez (2009). Plum et al. (2014) propose a new mathematical formulation based on a service formulation, where the set of all services S are defined beforehand. The set of services can handle several calls to the same port during the same route, i.e., it can include the non-simple routes, which better represents the services operated by liner shipping companies in the real world. In order to do that, the authors define the set of port calls B, and a service is defined to consist of a number of port calls. The model is defined with a series of continuous flow variables that represent the amount of demand that is transported on a certain port call leg of a service, among other flow variables which represent the flow of cargo from and to a specific port call in the different ports of the problem. This formulation allows the rejection of part of the demand, which is subtracted from the objective function incurring a penalty for not flowing the cargo. The model defines the decision variables in such a way that the flow of containers from one service to another can be considered correctly and transshipment can be modelled. Furthermore, the model imposes the services to have weekly frequency while limiting the fleet deployment according to the available fleet. Finally, the authors present an objective function where the operator's profit of the flowed cargo on the operating network is maximized, while the operational cost of the services and the cost for handling the cargo are minimized.

3.4 Outbound-inbound Principle with Transit Time Constraints

Wang and Meng (2014) incorporate the transit time constraints when designing the network in liner shipping problems. However, transshipment between services is excluded in this approach as they define the ship routes with the outbound-inbound principle. The problem is defined with a set R of geographical ship routes, which is an itinerary of port calls, using binary variables for the selection of these itineraries. The proposed model is a mixed-integer non-linear and non-convex programming model with an exponential number of decision variables, and it determines the network design and cargo routing of containers through the network. Binary variables are defined for assignment of arcs to the ship routes in order to construct feasible geographical ship routes. Furthermore, there exists a limited available fleet, and the fleet deployment is controlled by integer variables. Demand can be split among different vessels, and the model defines continuous variables for the amount of demand flowing through the arcs. These variables allow the model to define feasible patterns of the demand on the selected geographical ship routes. Additionally, binary variables are defined to ensure that the transit times of the cargoes are not violated. Finally, the model defines the port time as a function of the number of containers handled at the port. This is taken into account when the route length is enforced to have weekly frequencies. The problem is proved to be strongly NP-hard by reduction from the Bin Packing Problem, and Wang and Meng (2014) describe a column generation-based algorithmic scheme for its resolution. The approach efficiently finds high-quality solutions that can help planners to design better liner shipping networks.

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 commonplace 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 Benders' 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.

Let $G = (N, A)$ be a directed graph, where N represents the ports and A represents the arc set that connects the ports. Let K be the set of commodities with corresponding parameters as defined in Sect. 1.1. To each arc $(i, j) \in A$, further, define the corresponding cost, c_{ij}^k, of transporting one unit of commodity k through (i, j) and 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. Lastly, 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:

$$\min \quad \sum_{(i,j)\in A} \sum_{k\in K} c_{ij}^k x_{ij}^k \tag{9.5a}$$

$$s.t. \quad \sum_{(i,j)\in A:i=p} x_{ij}^k - \sum_{(i,j)\in A:j=p} x_{ij}^k = \xi_p^k \qquad p \in N, k \in K \tag{9.5b}$$

$$\sum_{k\in K} x_{ij}^k \le u_{ij} \qquad (i, j) \in A \tag{9.5c}$$

$$x_{ij}^k \ge 0 \qquad (i, j) \in A, k \in K. \tag{9.5d}$$

Here, the objective, (9.5a), is to minimize the total cost. Constraints (9.5b) are the flow conservation constraints, constraints (9.5c) are the capacity constraints, and constraints (9.5d) define the domain of the variables x_{ij}^k.

When fractional flows are allowed, the MCFP is solvable in polynomial time, but for larger instances, it is still computationally demanding. As it 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 constraint matrix and apply Dantzig-Wolfe decomposition (Ahuja et al. 1993; Karsten et al. 2015). First reformulate the problem to a path-flow formulation, where the goal is to allocate the commodities onto a number of flow paths from the commodity 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^k_a be the set of paths for commodity k, which uses the arc a. For each path, p, define its cost $c_p = \sum_{a \in A: p \in P^k_a} c^k_a$, and a corresponding decision variable f_p, deciding the flow through path p. The path-flow formulation can then be expressed as:

$$\min \sum_{k \in K} \sum_{p \in P^k} c_p f_p \tag{9.6a}$$

$$s.t. \sum_{p \in P^k} f_p = \xi^k_p \qquad k \in K \tag{9.6b}$$

$$\sum_{p \in \bigcup_{k \in K} P^k_a} f_p \leq u_a \qquad a \in A \tag{9.6c}$$

$$f_p \geq 0 \qquad k \in K, p \in P^k. \tag{9.6d}$$

The objective function, (9.6a), is to minimize the cost. Constraints (9.6b) ensure that all commodities are delivered and constraints (9.6c) assert that the arc capacity is not exceeded. Lastly, constraints (9.6d) define the domain of 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. This way, the path formulation can generally be solved faster than the arc formulation, described above. The path formulation makes it relatively easy to implement transit time constraints as they can be handled in the pricing problem.

Another efficient method of solving the MCFP (without time constraints) is by using so-called interior point methods, as is done by Álvarez (2009). In contrast to the simplex method, which searches through the vertices of the solution space, interior point methods search through solutions in its interior.

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 to calculate the objective value of a given solution, one has to solve the container flow problem. This makes the service selection problem difficult to solve optimally, 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* (SSSCRP) with a column generation and a Benders' 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 utilized to generate and choose new services for the vessels. As column generation solves only the LP relaxation of the problem, 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 Benders' 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 labeling 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 routes. The moves considered in the tabu-search for the services were deletion, change in vessel speed and change in number of vessels assigned. To guide the search, from the solution of the commodity flow problem, information about which services were under-/overutilized was used to increase/decrease the number of vessels and the speed. 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), which 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), in which in each iteration, an integer program is solved to update the current services.

Here follows a brief description of the algorithm from Brouer et al. (2015). Let $G = (N, A)$ be a complete directed graph, where N represents the ports and A represents the possible connections between ports. Let S denote the set of services, where each service, $s \in S$, visits a set of ports, $N_s \subseteq N$, and has a corresponding vessel class v_s, a number of assigned vessels m_s and a duration τ_s. The algorithm is initialized, 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. Let us define r^+_{is} (r^-_{is}) to be the estimated revenue change and t^+_{is} (t^-_{is}) to be the estimated duration change from including (excluding) port $i \in N$ in (from) service $s \in S$. Denote the weekly cost of using a vessel of the vessel class v_s by c_s and the number of free vessels of the type v_s by M_s. Lastly, let us define the binary variables λ_{is}^+ and λ_{is}^-, which control the inclusion and removal, respectively, of port i from service s, and the integer variable ω_s, which denotes the number of vessels to add to/subtract from service s. We also define a maximum number of inclusions, γ_s^+, and removals γ_s^- and a number of locksets L_i. For each service $s \in S$, we can then define the following mixed-integer program:

$$\max \quad \sum_{i \in N_s} r_{is}^+ \lambda_{is}^+ + \sum_{i \in N \setminus N_s} r_{is}^- \lambda_{is}^- - c_s \omega_s \tag{9.7a}$$

$$\text{s.t.} \quad \tau_s + \sum_{i \in N_s} t_{is}^+ \lambda_{is}^+ + \sum_{i \in F_s} t_{is}^- \lambda_{is}^- \leq 24 \cdot 7 \cdot (m_s + \omega_s) \tag{9.7b}$$

$$\omega_s \leq M_s \tag{9.7c}$$

$$\sum_{i \in N_s} \lambda_{is}^+ \leq \gamma_s^+ \tag{9.7d}$$

$$\sum_{i \in F_s} \lambda_{is}^- \leq \gamma_s^- \tag{9.7e}$$

$$\sum_{j \in L_i} \lambda_{js}^- \leq |L_i|(1 - \lambda_{is}^+) \qquad i \in N_s \tag{9.7f}$$

$$\sum_{j \in L_i} \lambda_{js}^- \leq |L_i|(1 - \lambda_{is}^-) \qquad i \in N \setminus N_s \tag{9.7g}$$

$$\lambda_{is}^+ \in \{0, 1\}, \quad i \in N_s \qquad \lambda_{is}^- \in \{0, 1\}, \quad i \in N \setminus N_s \qquad \omega_s \in \mathbb{Z}, \tag{9.7h}$$

where the objective (9.7a) is to maximize the increase in revenue. Constraint (9.7b) ensures that there are enough vessels assigned to keep the weekly frequency, and constraint (9.7c) specifies that no more than the number of free vessels can be added to the service. Constraints (9.7d) and (9.7e) set a limit on the number of insertions and removals and (9.7f) and (9.7g) enforce the locksets L_i. Constraints (9.7d), (9.7e), (9.7f) and (9.7g) 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 deteriorate 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, (9.7h) defines the domains 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, and 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, every tenth iteration the services with the lowest utilization are removed, and new services are created using the greedy construction 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 in advance by expert planners. 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-maximizing problem is denoted the Liner Service Planning (LSP) problem.

The transportation network consists of a set of ports N indexed by i and j and a set of candidate services S where each service $s \in S$ has N_s port calls. Associated with each candidate service $s \in S$ is a set of sailing arcs $a \in A_s$ where each arc represents the part of a ship's itinerary between two successive ports on the service route. The fleet is composed of several vessel classes and V denotes the set of these classes. There are M_v available vessels of each class $v \in V$, and for each service $s \in S$ we let m_v^s denote the required number of vessels of class $v \in V$. Associated with each service $s \in S$ is also a cost c_s and for each arc $a \in A_s$ a capacity g_a.

K denotes the set of commodities where an origin port o_k, a destination port d_k and a demand δ_k are associated with each commodity $k \in K$. It is allowed to split the flow of each commodity, and a penalty cost c_k^r per container is used to penalize rejected demand of commodity k.

Given a commodity's route, a *sub-path* is defined as the part of the route in which the container travels 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 modeling approach falls somewhere between the two more traditional modeling approaches of either using arc-flow, i.e. over sailing edges, or path-flows, i.e. 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, h_k denote the maximum allowed number of sub-paths on which commodity k can travel. Note that h_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 hth stage is described by the variable x_{ijs}^{hk} for $s \in S$, $\langle i, j, s\rangle \in A_s$ and $h = 1, 2, \ldots, h_k$. 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:

$$\min \sum_{s\in S} c_s y_s + \sum_{k\in K}\sum_{s\in S}\sum_{h=1}^{h_k} c_{ijs}^k x_{ijs}^{hk} + \sum_{k\in K} c_k^r z_k \tag{9.8a}$$

$$s.t. \sum_{s\in S}\sum_{\langle o_k,j,s\rangle\in H_s} x_{o_k js}^{1k} + z_k = \delta_k \qquad \forall k \in K, \tag{9.8b}$$

$$\sum_{s\in S}\sum_{i:\langle i,j,s\rangle\in H_s} x_{ijs}^{hk} - \sum_{s\in S}\sum_{l:\langle j,l,s\rangle\in H_s} x_{jls}^{h+1,k} = 0 \qquad \forall k \in K, j \in N \setminus \{o_k, d_k\}, h = 1, \ldots, h_k - 1, \tag{9.8c}$$

$$\sum_{k\in K}\sum_{h=1}^{h_k}\sum_{\langle i,j,s\rangle\in H_s : a\in A_{ij}^s} x_{ijs}^{hk} \le g_a y_s \qquad \forall s \in S, a \in A_s \tag{9.8d}$$

$$\sum_{s\in S} m_v^s y_s \le M_v \qquad \forall v \in V, \tag{9.8e}$$

$$x_{ijs}^{hk} \ge 0 \qquad \forall k \in K, s \in S, \langle i, j, s\rangle \in H_s, h = 1, \ldots, h_k, \tag{9.8f}$$

$$z_k \ge 0 \qquad \forall k \in K, \tag{9.8g}$$

$$y_s \in \{0, 1\} \qquad \forall s \in S. \tag{9.8h}$$

The objective function (9.8a) minimizes 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 minimization problem as opposed to a profit maximization problem where c_k^r would instead represent the revenue for transporting one unit of commodity k.

Constraints (9.8b) ensure that the flow of each commodity k is assigned to sub-paths incident to the corresponding origin port o_k. They also ensure that this 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 (9.8c) are flow-balancing constraints for intermediate ports. Together with constraints (9.8b), these constraints ensure that for each commodity k the demand subtracted any unmet demand will arrive at the destination port using at most h_k sub-paths, i.e. fulfilling the constraint on a maximum number of transshipments.

Constraints (9.8d) impose capacity constraints on the sailing arcs and ensure that only sub-paths from the selected services can be used. Constraints (9.8e) ensure that no more than the available vessels are used. Finally, constraints (9.8f), (9.8g) and (9.8h) 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 Optimization-Based Heuristic Procedure

Balakrishnan and Karsten (2017) propose an optimization-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 constraint, 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 was 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 a 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.

In order to find the backbone flow, an artificial network $G = (N, A)$ is used where N is the set of ports and A is a complete, directed graph. 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 the edge. This can be expressed as a concave function $f(x)$ 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 Fig. 9.2 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 optimize the container flow.

Let the set of commodities K and demands ξ_i^k be defined as in (9.1), and 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

$$\min \quad \sum_{(i,j)\in A} f(\sum_{k\in K} x_{ij}^k) \tag{9.9a}$$

$$s.t. \quad \sum_{(i,j)\in A} x_{ij}^k - \sum_{(j,i)\in A} x_{ji}^k = \xi_i^k \qquad i \in N, k \in K \tag{9.9b}$$

$$x_{ij}^k \geq 0 \qquad (i, j) \in A, k \in K. \tag{9.9c}$$

As before, the objective, (9.9a), is to minimize the total cost, and constraints (9.9b) are the flow conservation constraints. Constraints (9.9c) define the domain of the variables.

Since the model is non-linear, (Krogsgaard et al. 2018) solve the problem heuristically through a randomized 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 flown. Generally, the first containers are more decisive for the arcs used heavily in the final solution than the last containers flown. 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 Fig. 9.4. The figure clearly shows that only a fraction of the possible arcs is used in the solution.

6.1 Greedy Heuristic for Generating Services

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 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

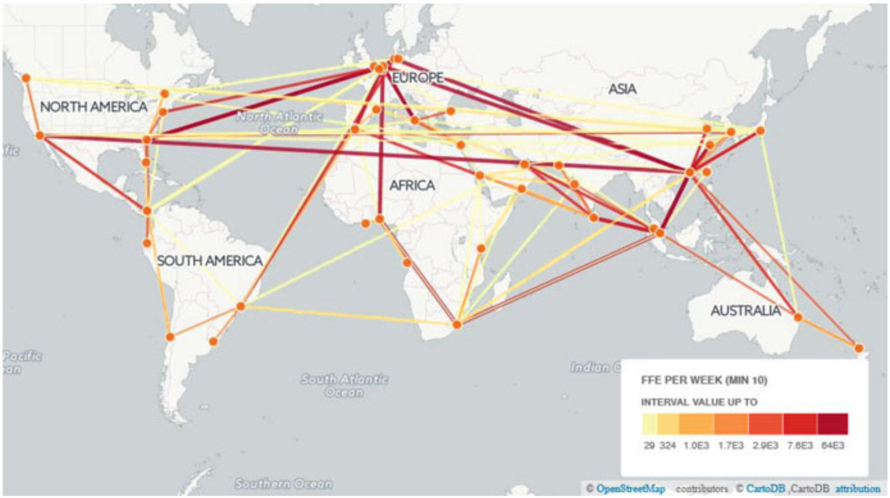

Fig. 9.4 Typical backbone flow for the *WorldSmall* instance (Source: Krogsgaard et al. 2018)

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 traveling 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 reasonable 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 s and results in profitable solution, although the resulting network is far from optimal.

6.2 *Network Optimization*

In order to improve the initial services found by the greedy heuristic, Krogsgaard et al. (2018) use a Variable Neighborhood 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 neighborhood structures throughout the search to exploit the benefits from neighborhood changes. When a local optimum is encountered, it is escaped by doing a random move, a `shake`, from the best known solution and do hill climbing from here until a new local optimum is reached. If this solution is better than the previously best known, the search is continued from here with a new shake; otherwise the search returns to the previously best known solution and searches from here again after a new shake. The pseudocode of the metaheuristic is given in Algorithm 1.

Algorithm 1 Improvement Algorithm

Initialisation: Find an initial solution x
while stopping criterion not met **do**
    generate a new solution x' from x (`shake`)
    while any neighbourhood in N is unused (`local search`) **do**
        choose at random an unused neighbourhood and search from x'
        if an improved solution x'' is found **then**
            Set $x' := x''$ and set all neighbourhoods unused
        if x' is better than x (`test solution`) **then**
            Set $x := x'$
return x

As can be seen from the pseudocode, the `local search` procedure terminates after all neighborhoods have been tested without yielding an improving solution, as a local optimum with respect to all neighborhoods must then have been encountered. The `shake` procedure is applied less frequently than in a standard VNS framework. A lower degree of randomness is preferable here, because the evaluations are relatively expensive. It is thus desirable to search directly for a local optimum with respect to all neighborhoods before randomly altering the solution.

Although the local search only accepts moves that have an expected improvement, some moves may turn out to be degrading when calculating the real objective function. These moves are nevertheless kept on to progress the search. This can, however, lead to cycling in the local search, as it might both be expected to be an improvement to first insert a port and to remove it afterward. To break such cycles, only 20 loops are allowed in the `local search` part, after which the algorithm must continue to `test solution`.

If a cycle is encountered or a local optimum has been reached, the `shake` procedure is applied to progress the search from another point in the solution space. The procedure must change the solution sufficiently to escape the local optimum, but should, on the other hand, not destroy good characteristics of the solution. Preliminary studies show that there is a high risk of changing the solution too much to be able to return to a good solution, and a relatively modest shake procedure is thus implemented. This procedure modifies a number of services by either inserting or removing a port randomly, without considering the effect on objective value. To avoid inserting an obviously irrelevant port, a distance requirement is enforced such that only ports relatively close to the service can be inserted. The number of modified services is 10% of the total number of services and a least one.

In each iteration of `local search`, a neighborhood is randomly selected, and one or more services are altered through that neighborhood. Six different neighborhoods are applied: *Insert port*, *Service omission*, *Service unserved port*, *Remove port*, *Simple remove port* and *Create feeder services*.

In order to select the best move, delta evaluation is used to avoid time-consuming evaluations of the multi-commodity flow for the entire graph. Instead, a small graph (*rotation graph*) is constructed, covering only the rotation currently being altered. As this graph is much faster to evaluate, more moves can be tested before one is selected for implementation. See Krogsgaard et al. (2018) for a detailed description of the neighborhoods.

Promising computational results using the LINER-LIB instances are reported in Krogsgaard et al. (2018). The *WorldLarge* instance can be solved within 1 hour, while the smaller instances have much tighter CPU time limits. The authors report that they can improve the solutions of Brouer et al. (2014a,b), for instances *WestAfrica*, *WorldSmall* and *WorldLarge*. Perhaps the most important result is that the number of transshipments in general is very low, being below 1.14 per commodity. For the smaller instances, the number of transshipments is below 0.5 per commodity. Fewer transshipments mean shorter port stays, and hence vessels are not as likely to be restricted by the maximum transit times.

7 Speed Optimization

As described in Brouer et al. (2017), a key tool in achieving lower fuel consumption in liner shipping is to reduce the sailing speed between the serviced ports. However, a lower sailing speed will increase the transit times for containers, and more vessels are needed to transport the same amount of cargo as it takes longer time to complete a rotation.

Bunker consumption for a vessel profile is often modelled as a cubic function of speed, but in practice it depends not only on the speed of operation but also on wind and currents, the vessel type, the draft of the vessel, the time since the hull was cleaned and the number of reefer containers powered by the vessel's engine. During a round trip, the vessel may sail at different speeds between ports. The vessel may *slow steam* to save bunker fuel or increase speed to meet a crucial transit time. Hence speed optimization is a complex trade-off between these two criteria. A good strategy is to speed up when the vessel is fully loaded (and hence many containers need to meet their transit time), while slow steaming can be used when the load is low.

Several recent papers study speed optimization with increasing complexity and integration with routing decisions. Ronen (2011) presents a simple model where the speed and number of allocated vessels are optimized to minimize cost on a single predefined service and a single speed for the full service is assumed. Meng and Wang (2011a) also work with a single service but use a more detailed model, taking, for example, transit times into account. Further, the speed is optimized for each individual leg. Wang and Meng (2012) consider a liner shipping network with multiple predefined services and present a non-linear MIP model to optimize the ship deployment and speed of those services and the container routing through this network. The sailing speeds are optimized on each leg individually. In this model, however, no transit time constraints are considered. Kim (2013) presents a model to determine the speed and bunkering ports for a single vessel on a predetermined path. There are no transit time constraints, but a cost is imposed for each day a container is on the ship.

Another appearance of speed optimization is in the paper by Álvarez (2009), in which it is integrated with liner shipping network design. However, the model assumes a single speed for the full service, and the model has no transit time constraints.

Reinhardt et al. (2016) present a model for speed optimization of an existing liner shipping network which adjusts berthing times to minimize the overall bunker consumption. It is assumed that all services, as well as the number of deployed vessels, are fixed and that all containers are flowed along the same route as before speed optimization. Moreover, transit time constraints are taken into account. When rescheduling the berthing times, the overall transit time of a demand may change. Hence a constraint is imposed for each commodity ensuring that the transit time is within an acceptable range. A penalty is paid for each change in port calls to keep the schedule similar to the original one. Reinhardt et al. (2016) report that the model is able to save around 2% of bunker consumption while keeping all transit times unchanged. If transit times can be extended by up to 48 h, a saving of around 6–7% can be achieved.

Karsten et al. (2018) present a more advanced speed optimization model, where the services are fixed, but the speed on each leg is allowed to vary, and hence commodities may take a different route if speed changes allow for a cheaper or faster route than currently available. The problem is solved using Benders' decomposition, and results indicate that the flow changes significantly when the speed on the individual legs is changed.

Finally, Karsten et al. (2017a) consider a complete network design problem with speed optimization on individual legs, by extending the matheuristic from Karsten et al. (2017b). The leg speeds are iteratively calculated for each single service based on the current flow of containers. The method adjusts speed to the required transit times of the current container routings throughout the round trip. The individual leg speeds are calculated by solving a MIP model with the objective of minimizing the bunker consumption. A piece-wise linear function is used to approximate the cubic bunker consumption function.

8 Conclusion

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

Slow steaming together with larger vessels has proven to be an efficient tool for reducing 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. Bigger 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 the port stays. It is therefore necessary to design routes such that fewer transshipments are needed while still ensuring a good utilization of the mega-vessels.

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.

In the future we will see container vessels operating with new, greener, propulsion types. Electric vessels may operate shorter routes, while liquid natural gas (LNG) may be used for operating longer routes. The new propulsion types will make it necessary to completely rethink route net design, since refueling/recharging will be more complicated, and vessels will have a more limited range of operation.

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. (2014a) and Li et al. (2015), but 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.

References

Agarwal, R., & Ergun, Ö. (2008). Ship scheduling and network design for cargo routing in liner shipping. *Transportation Science, 42*(2), 175–196.

Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network flows: Theory, algorithms, and applications*. Englewood Cliffs: Prentice Hall.

Álvarez, J. F. (2009). Joint routing and deployment of a fleet of container vessels. *Maritime Economics & Logistics, 11*(2), 186–208.

Archetti, C., & Speranza, M. G. (2014). A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization, 2*(4), 223–246.

Balakrishnan, A., & Karsten, C. V. (2017). Container shipping service selection and cargo routing with transshipment limits. *European Journal of Operational Research, 263*(2), 652–663.

Brouer, B. D., Álvarez, J. F., Plum, C. E. M., Pisinger, D., & Sigurd, M. M. (2014a). A base integer programming model and benchmark suite for liner-shipping network design. *Transportation Science, 48*(2), 281–312.

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

Brouer, B. D., Desaulniers, G., Karsten, C. V., & Pisinger, D. (2015). A matheuristic for the liner shipping network design problem with transit time restrictions. In F. Corman, S. Voß, & R. R. Negenborn (Eds.), *Computational logistics* (pp. 195–208). Cham: Springer.

Brouer, B. D., Karsten, C. V., & Pisinger, D. (2016). Big data optimization in maritime logistics. In A. Emrouznejad (Ed.), *Big data optimization: Recent developments and challenges* (Studies in big data, Vol. 18, pp. 319–344). Cham: Springer.

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

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

Christiansen, M., Fagerholt, K., Nygreen, B., & Ronen, D. (2013). Ship routing and scheduling in the new millennium. *European Journal of Operational Research, 228*(3), 467–483.

Dithmer, P., Reinhardt, L. B., & Kontovas, C. A. (2017). The liner shipping routing and scheduling problem under environmental considerations: The case of emissions control areas. In *International Conference on Computational Logistics* (pp. 336–350). Springer.

Fagerholt, K., Gausel, N. T., Rakke, J. G., & Psaraftis, H. N. (2015). Maritime routing and speed optimization with emission control areas. *Transportation Research Part C: Emerging Technologies, 52*, 57–73.

Germanischer Lloyd. (2013). http://www.balkanlloyd.com/news/96-germanischer-lloyd-has-conducted-research-showing-that-new-and-efficient-4-500-teu-panamax. [Online; Accessed 20 Feb 2017].

Hansen, P., & Mladenovic, N. (2014). Variable neighborhood search. In E. K. Burke & G. Kendall (Eds.), *Search methodologies – Introductory tutorials in optimization and decision support techniques* (pp. 313–337). New York: Springer.

IMO. (2014). Third imo greenhouse gas study 2014. Technical report, International Maritime Organization.

Karsten, C. V. (2015). *Competitive liner shipping network design*.Ph. D. Thesis, DTU Management Engineering.

Karsten, C. V., Pisinger, D., Røpke, S., & Brouer, B. D. (2015). 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.

Karsten, C. V., Brouer, B. D., & Pisinger, D. (2017a). Competitive liner shipping network design. *Computers & Operations Research, 87*, 125–136.

Karsten, C. V., Brouer, B. D., Desaulniers, G., & Pisinger, D. (2017b). Time constrained liner shipping network design. *Transportation Research. Part E: Logistics and Transportation Review, 105*, 152–162.

Karsten, C. V., Røpke, S., & Pisinger, D. (2018). Simultaneous optimization of container ship sailing speed and container routing with transit time restrictions. *Transportation Science, 52*(4), 739–1034.

Kim, H. J. (2013). A lagrangian heuristic for determining the speed and bunkering port of a ship. *Journal of the Operational Research Society, 65*(5), 747–754.

Kontovas, C. A. (2014). The green ship routing and scheduling problem (GSRSP): A conceptual approach. *Transportation Research Part D: Transport and Environment, 31*, 61–69.

Krogsgaard, A., Pisinger, D., & Thorsen, J. (2018). A flow-first route-next heuristic for liner shipping network design. *Transportation Science, 72*(3), 358–381. Special Issue on Emerging Challenges in Transportation Planning.

Lee, C.-Y., & Song, D.-P. (2017). Ocean container transport in global supply chains: Overview and research opportunities. *Transportation Research Part B: Methodological, 95*, 442–474.

Li, C., Qi, X., & Lee, C.-Y. (2015). Disruption recovery for a vessel in liner shipping. *Transportation Science, 49*(4), 900–921.

Maersk. (2017). The world's largest ship. https://www.maersk.com/en/explore/fleet/triple-e/environment

Meng, Q., & Wang, S. (2011a). Optimal operating strategy for a long-haul liner service route. *European Journal of Operational Research, 215*(1), 105–114.

Meng, Q., & Wang, S. (2011b). Liner shipping service network design with empty container repositioning. *Transportation Research Part E: Logistics and Transportation Review, 47*(5), 695–708.

Meng, Q., Wang, S., Andersson, H., & Thun, K. (2014). Containership routing and scheduling in liner shipping: Overview and future research directions. *Transportation Science, 48*, 265–280.

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

Montreuil, B. (2011). Towards a physical internet: Meeting the global logisitcs sustainability grand challenge. *Logistics Research, 3*, 71–87.

Neamatian Monemi, R., & Gelareh, S. (2017). Network design, fleet deployment and empty repositioning in liner shipping. *Transportation Research Part E, 108*, 60–79.

Notteboom, T. E. (2006). The time factor in liner shipping services. *Maritime Economics & Logistics, 8*(1), 19–39.

Notteboom, T. E., & Vernimmen, B. (2009). The effect of high fuel costs on liner service configuration in container shipping. *Journal of Transport Geography, 17*,(5), 325–337.

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

Psaraftis, H. N., & Kontovas, C. A. (2013). Speed models for energy-efficient maritime transportation: A taxonomy and survey. *Transportation Research Part C: Emerging Technologies, 26*, 331–351.

Psaraftis, H. N., & Kontovas, C. A. (2015). Slow steaming in maritime transportation: Fundamentals, trade-offs, and decision models. In C.-Y. Lee & Q. Meng (Eds.), *Handbook of ocean container transport logistics: Making global supply chains effective* (pp. 315–358). Heidelberg: Springer.

Reinhardt, L. B., & Pisinger, D. (2012). A branch and cut algorithm for the container shipping network design problem. *Flexible Services and Manufacturing Journal, 24*(3), 349–374.

Reinhardt, L. B., Plum, C. E. M., Pisinger, D., Sigurd, M. M., & Vial, G. T. P. (2016). The liner shipping berth scheduling problem with transit times. *Transportation Research Part E: Logistics and Transportation Review, 86*, 116–128.

Ronen, D. (1983). Cargo ships routing and scheduling: Survey of models and problems. *European Journal of Operational Research, 12*(2), 119–126.

Ronen, D. (1993). Ship scheduling: The last decade. *European Journal of Operational Research, 71*(3), 325–333.

Ronen, D. (2011). The effect of oil price on containership speed and fleet size. *Journal of the Operational Research Society, 62*(1), 211–216.

Stopford, M. (2009). *Maritime economics (3rd ed.)*. London: Taylor & Francis Ltd. ISBN:9780415275583.

Sun, Z., & Zheng, J. (2016). Finding potential hub locations for liner shipping. *Transportation Research Part B: Methodological, 93*, 750–761.

Thun, K., Andersson, H., & Christiansen, M. (2017). Analyzing complex service structures in liner shipping network design. *Flexible Services and Manufacturing Journal, 29*(3–4), 535–552.

Toth, P., & Vigo, D. (2015). *Vehicle routing: Problems, methods and applications* (Vol. 18). Philadelphia: SIAM.

Unctad. (2017a). Review of maritime transport. Technical report, United Nations Conference on Trade and Development.

Unctad. (2017b). UnctadSTAT. http://unctadstat.unctad.org/wds/TableViewer/tableView.aspx. [Online; Accessed 9 Feb 2017].

Wan, Z., Zhu, M., Chen, S., & Sperling, D. (2016). Pollution: Three steps to a green shipping industry. *Nature, 530*, 275.

Wang, S., & Meng, Q. (2012). Sailing speed optimization for container ships in a liner shipping network. *Transportation Research Part E: Logistics and Transportation Review, 48*(3), 701–714.

Wang, S., & Meng, Q. (2014). Liner shipping network design with deadlines. *Computers & Operations Research, 41*, 140–149.