



A service flow model for the liner shipping network design problem

Plum, Christian Edinger Munk; Pisinger, David; Sigurd, Mikkel M.

Published in:
European Journal of Operational Research

Link to article, DOI:
[10.1016/j.ejor.2013.10.057](https://doi.org/10.1016/j.ejor.2013.10.057)

Publication date:
2014

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
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. <https://doi.org/10.1016/j.ejor.2013.10.057>

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.

Chapter 5

A Service Flow Model for the Liner Shipping Network Design Problem

Christian E. M. Plum^{*}, [‡] David Pisinger^{*} Mikkel M. Sigurd[‡]

^{*}Department of Management Engineering, Technical University of Denmark,
Produktionstorvet, Building 426, DK-2800 Kgs. Lyngby, Denmark
cemp@man.dtu.dk, pisinger@man.dtu.dk

[‡]Maersk Line
Esplanaden 50, DK-1098 København K.
mikkel.muhldorff.sigurd@maersk.com

1

Abstract Global liner shipping is a very competitive industry, requiring liner carriers to carefully deploy their vessels efficiently, to construct a cost competitive network. This paper presents a novel compact formulation of the liner shipping network design problem (LSNDP) based on service flows. The formulation alleviates issues faced by arc flow formulations with regards to handling multiple calls to the same port, which has not been fully dealt with earlier by LSNDP formulations. This is done by introducing service nodes, together with port nodes in a graph of the problem. Arcs from a port node to a service node represent whether a service is calling some port, and what demand loads and unloads at the port call. This representation allows recurrent calls of a service to the a port, which other optimal LSNDP models have not handled. The model ensures strictly weekly frequencies of services, ensures that port-vessel draft capabilities are not violated, respects vessel capacities and number of vessels available. The profit of the generated network is maximized, i.e. the revenue of flowed cargo subtracted a penalty for not flowed cargo and operational costs of the network. The model can be used to design liner shipping networks to utilize a container carriers assets efficiently and to investigate possible scenarios of changed market conditions. The model is solved as a Mixed Integer Program. Results are presented for the two smallest instances of the benchmark suite LINER-LIB 2012 presented in Brouer et al. [5].

Keywords Liner shipping, Network design, Maritime optimization

¹Submitted for a special issue in *European Journal of Operations Research* on Maritime Logistics

5.1 Introduction

When manufactured goods are transported from one corner of the world to another it is likely to happen in a container. These containers are carried by up to 400 meter long container vessels carrying tens of thousands of containers. Liner shipping carriers operating these vast vessels construct intricate networks of shipping routes that in their interaction provide fast and, relative to any other transport mode, cheap transport that operate at the core of the worlds supply chains.

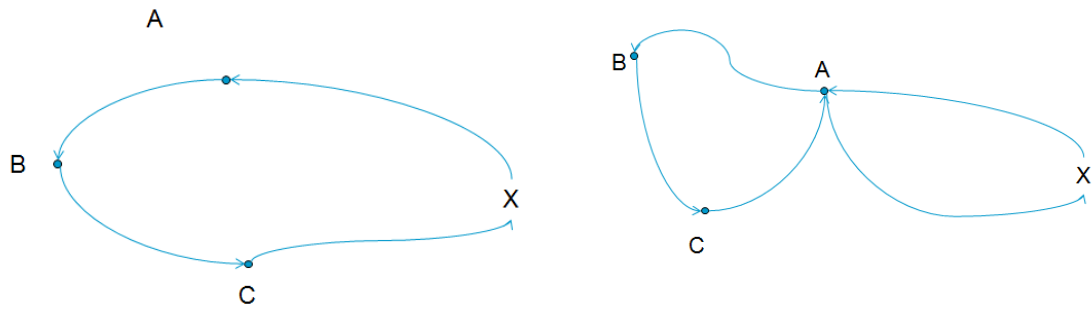
A global container shipping network is extremely costly to operate, with Maersk Line using a two-digit billion USD amount yearly to maintain this. Therefore, even a small improvement of the network's utilization, costs, service levels, etc. can have a significant impact. At the same time the cost structure of the network can be very volatile; by developing models that can investigate an increased cost or reduced demand, the network can rapidly be modified to adapt for these changed market conditions.

The basic cost components of a container shipping network are: vessel costs, bunker fuel costs, port call fees and container move costs in ports. These have been very volatile in the past years, with fuel costs reaching record levels in 2008, to fall again during the financial crisis, followed by a rise in prices during the recovery bun [2]. Vessel charter rates are similarly fluctuating and have fallen dramatically as the financial crisis hit world trade tcc [1]. Furthermore, there is an increasing interest in limiting CO2 emissions, which is related to the fuel consumption. This is already an important aspect of a shipping line company's public profile, which is expected to have increased focus in the years to come. All these factors add to the importance of liner shipping network design. This paper presents a novel model of LSNDP. A liner shipping network consists of a number of *services*, each service being sailed by a fixed number of vessels following the same roundtrip, much like a bus transit network. These services are able to tranship containers between each other at ports. Each port to port sailing by a service is denoted the service's *legs*. Each port call on the roundtrip will be served at a fixed frequency predominantly weekly (as a bus route served every 20'th minute) by a vessel. To utilize vessels best possible, service roundtrip times are thus generally a multiplier of 7 days.

These services constitute the network, that the *demand* must be transported on. Each demand is a fixed weekly volume of containers requiring transport from a specified origin to a specified destination. The demand will pay a revenue for being served, but it is acceptable to leave some demand at the cost of paying a goodwill penalty. The path of the demand will incur additional cost for moving containers on and off the vessels. The goal is then to construct a liner shipping network, consisting of some services, allowing for the transport of containers, with the aim of maximizing the profit of operating the network, e.g. the revenue subtracting costs for services, move cost and goodwill penalties.

In the proposed model, services are constructed by opening links from each service (the number of considered services being fixed) to a number of ports, using specialized service-port arcs. This can be reconstructed to a service traversing between the ports, using auxiliary variables. Likewise the demands will flow from origin port to a service and from a service to destination port (or transshipment), on specialized service-port arcs. The specific port to port path can be reconstructed using auxiliary variables. As compared to LSNDP models of related problems this model has the advantages of allowing non-simple cycles with any number of calls to one or more *butterfly ports*. Services with one or more butterfly ports is often denoted *butterfly routes*, this approach is used in service design for different reasons:

- Increased capacity on the legs between the port calls, as the service can carry less cargo on these legs. This capacity can be used for other cargo. E. g. in Figure 5.1(b) more capacity will be available between the first and second call to port A, as exports from A to region X are not carried here. An example could be Singapore as a butterfly port on a Europe to North Asia / Japan service.
- Two services with non-weekly frequency can be combined to a service with weekly frequency. E.g. Let D, E, F be ports, then service 1 calling ($D \rightarrow E \rightarrow D$) with a 4 day roundtrip time



(a) A part of a service departing from a region X, calling ports A, B and C once each, then returning to region X.

(b) A part of a service departing from a region X, calling ports A, B, C and A again, then returning to region X. A Butterfly service with A as a *butterfly port*.

Figure 5.1: Examples of two services with and without a *butterfly port*.

and service 2 ($D \rightarrow F \rightarrow D$) with a 10 day roundtrip time, can be combined to a service 3 ($D \rightarrow E \rightarrow D \rightarrow F \rightarrow D$) with a 14 day roundtrip time and thus satisfying the weekly frequency requirement, by allowing double calls at port D.

- Draft limits at later ports may require that the vessel is eased before port call, a double port call will alleviate this on port calls in between.
- Improving transit time, as the extra port call will allow for faster imports or exports to remaining ports on the service. E. g. in Figure 5.1(b) exports from port A to region X will be faster with double calls.

Note that draft dependent on vessel load and transit time is not considered in the model, but still it is a cause for using butterfly routes. Thus it is important for a LSNDP model, to accurately model butterfly port calls.

The problem of LSNDP is a strategic one, where network design decisions are made on a 6-12 months horizon, subject to amendments at a later stage. A concern with the LSNDP is that all demand is considered deterministic, with a fixed demand between the worlds ports every week. This assumption is far from reality, as container demand is subject to large fluctuations from week to week. However, a deterministic approach is still relevant, as the demand on an aggregated level gives a more stable picture. E.g. when looking at demand from continent to continent, the forecasts will be more accurate. A deterministically designed network can to some degree be corrected by short term amendments to vessel schedules, using third party tonnage or rolling cargo to the next sailing. An alternate approach to solve liner shipping network design problems could be a stochastic approach. However, stochastic models generally greatly add to the complexity compared to deterministic models. As the deterministic version of LSNDP is very hard to solve, it is unlikely that a stochastic approach will scale to anything reasonable, hence this approach has not been investigated.

This *service flow model for the liner shipping network design problem* will be abbreviated SFM.

The model has been implemented as a MIP, solved by a commercial solver. Results are reported on data from the LSNDP Benchmark instances LINER-LIB 2012, [5].

5.2 Literature

In this section we summarize the literature which has been used directly in our work and is closely related to the considered problem. For a general introduction to the maritime industry please refer to Stopford [13]. For a review of research in maritime optimization refer to Christiansen et al. [6]

and Christiansen et al. [7]. For a detailed introduction to liner shipping network design and details on the structure of the liner shipping business, its operational requirements and cost structure refer to Brouer et al. [5]. A recent overview of research in container routing and scheduling problems in liner shipping is given by Meng et al. [10].

Work on liner shipping network design is particularly relevant for this paper. The work of Shintani et al. [12] has a detailed description of the cost structure of LSNDP and includes consideration of repositioning empty containers. A service is constructed, assuming all demand is flowed if a port is called. The method allows for butterfly ports, by using duplicate virtual ports, but only for a single service.

Two decomposition methods was presented by Agarwal and Ergun [3], a bender's and a column generation based algorithm. The column generation algorithm considers the flows of the containers in a master problem, and the pricing problem considers the generation of vessel rotations by detecting negative cost cycles. Multiple services are generated and transshipment is possible. These algorithms scale well to large instances, but transshipment costs are excluded. The methods are based on a time space graph, which allows for multiple visits to a port as long as the visits do not happen on the same day of the week.

Alvarez [4] proposes a heuristic column generation based method for solving the joint routing and deployment of a fleet of container vessels. The method also considers flow in the master problem and run / service generation in the sub problem. The method is able to solve instances with 7 ports. A solution for a 120 ports instance is reported, using more aggressive heuristic approaches. The algorithm, allows butterfly ports, but can not actually handle butterfly loops, incorrectly calculating towards capacity or transshipment cost as discussed in section 5.3.1.

Reinhardt and Pisinger [11] propose a branch and cut method for the LSNDP. The formulation allows pseudo-simple cycles, meaning a cycle with a single butterfly port with at most two calls. A clover cut for the problem is derived, and some cuts from VRP are also used. This gives a branch and cut algorithm, which outperforms a pure MIP-model solved by CPLEX. Instances with up to 10 ports are solved optimally.

The work of Brouer et al. [5] describes the domain of LSNDP, discusses the relevant scoping, proposes a model of the problem, and presents a number of benchmark instances denoted LINER-LIB 2012, for the LSNDP based on real world problems. A heuristic column generation based algorithm is presented in Brouer et al. [5], which solves a number of the LINER-LIB 2012 instances. The algorithm, which can be seen as an extension of Alvarez [4], have fixed the butterfly issues, but only allows a single butterfly port per service.

Other research in liner shipping network design is Gelareh and Pisinger [8] who proposes a hub and spoke model for which a primal decomposition based solution method is derived. The method does not allow for butterfly ports. A heuristic algorithm for a scheduling and container routing problem is presented by Wang and Meng [14], but does not design the rotations. A formulation considering empty container repositioning, which considers a set of input shipping lines is found in Meng and Wang [9] and dealing with robust schedule design in Wang and Meng [15]. None of these caters for the generation of new services.

To the best of the authors knowledge no model or algorithm have been developed considering transshipment costs, allowing any number of butterfly ports on a service, and none that allows the butterfly ports to have more than two calls. This paper considers this by allowing any number of butterfly ports, with any number of port calls, while considering transshipment costs.

Overview of the paper In section 5.3 we introduce the mathematical notation used in the paper and explain details on the problem structure. The concept of service flow is central for this paper and explained in detail in section 5.3.1. This is used to formulate a model for the LSNDP in section 5.3.2. This MIP model has been implemented and solved with a commercial solver. Computational results are reported in section 5.4, with the key contributions of providing solutions for two instances of LINER-LIB 2012, while considering operational constraints as capacity, port draft and butterfly ports, which no method previously has. We conclude on the paper in section 5.5 and discuss directions for future research.

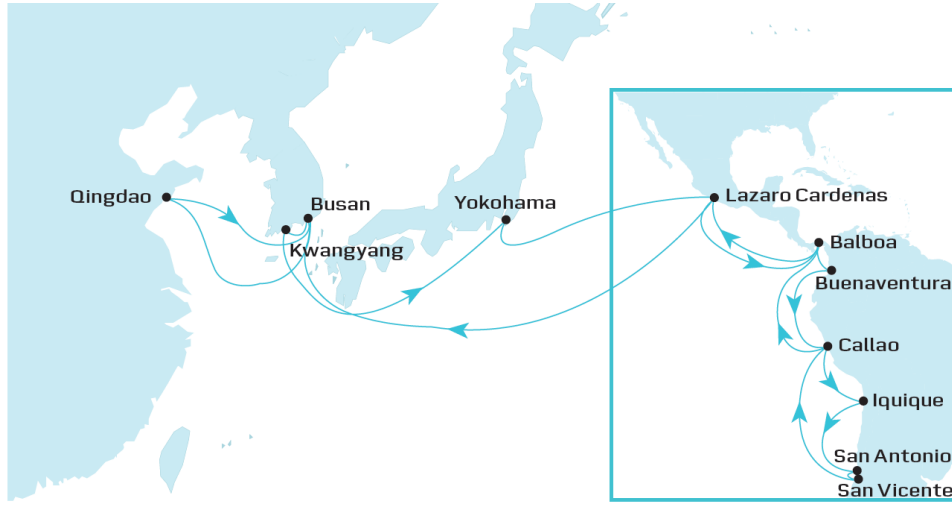


Figure 5.2: The AC3 Service, transporting containers between north Asia and central and south Americas. Note that the ports Busan, Lazaro Cardenas, Balboa and Callao have butterfly calls.

5.3 Mathematical Formulation

The notation used throughout the paper is introduced in this section. Vessels are available in different vessel classes, $v \in V$, characterized by their forty foot container capacity c_v ; a maximal number z_v of available vessels of the vessel class v . The ports $p \in P$ have the following characteristics: a port call cost for a vessel class is $w_{p,v}$; move cost for a container is u_p . The set of directed edges considered, $a \in A \subseteq P \times P$, is a subset of the complete graph spanned by the ports, let $a(i, j)$ denote the arc from port i to port j . All distances $d_{ij} = d_a$ are assumed to satisfy the triangle inequality. $o(a)$ and $d(a)$ respectively is the head and tail ports of the arc. A distance d_a from $o(a)$ to $d(a)$ is assumed not to travel through canals subject to fee's for passing (i.e. Suez and Panama). These canals are assumed to exist in $p \in P$ with appropriate $w_{p,v}$, so a canal traversal will only be possible by visiting this port and thus paying the fee.

The demand $k \in K$ with volume F^k , where $o(k)$ and $d(k)$ (same notation as for arcs) respectively is origin and destination port of the demand, is assumed to flow continuously over the considered time-period, with a steady supply every week. Demand is only to be satisfied if profitable, the revenue for doing so is q^k per container. We have a goodwill penalty for not flowing a demand \tilde{q}^k .

A service $s \in S$ is a directed, possibly non-simple, cycle of ports $p \in P$, traversed by a given vessel class v being intrinsic to s . The port calls of s , can be denoted the *rotation* of s . This rotation is traversed by an integer number τ_s of identical vessels of a vessel class v calling each port in the rotation with a weekly frequency. Services are required to have weekly frequency, i.e. they must follow a fixed schedule with total roundtrip time a multiple of 7 days, i.e. $7 \cdot \tau_s$. Let $s \in v$ imply that s is sailed by vessel class v and $v(s)$ indicate the vessel class used by service s .

A service consists of a number of port calls $b \in B = \{1, \dots, |B|\}$. We will also refer to the b 'th leg of service s , which will be the sailing from the b 'th port call to the $b + 1$ 'th port call. Note that, as the call legs b are cyclic, then $b' = |B| + 1 = 1$ to enforce a cyclic service. A service can have no more port calls than $|B|$.

Vessels are assumed to sail at their design speed s_v^v , measured in nautical miles per hour, during their whole rotation, and port stays are fixed to 24 hours for all port calls, as done in LINER-LIB 2012. These are both very broad assumptions based roughly on actual services, but with large variations. The bunker consumption at speed s_v^v is f_v^v , and the additional idling consumption of the vessel f_0^v is used irrespective of whether the vessel is sailing, for electricity, pumping and other purposes. Both consumption types are measured in metric tonnes per day.

The demand k is then transported on one of these services s either in a **direct path**, where s calls $o(k)$ loads the cargo and carries it all the way to $d(k)$ and unloads the demand. The demand

can also be carried in a **transshipment path**, where a service, s_1 calls port $o(k)$ loads the cargo and carries it to some transshipment port t where the demand is unloaded and loaded by a different service, s_2 carrying it to $d(k)$. A path can involve multiple such transshipments.

Both the demand (weekly number of containers to be flowed) and supply (number of vessels to sail on a service to achieve weekly frequency) is defined in weekly terms, hence the time dimension does not need explicit representation in the problem graph, which can be seen as a recurring representation of the demand / supply balance over a week. The model will then aim to install weekly capacity in the network on which the weekly demand can flow.

This modeling follows industry practice, where all larger and most smaller services have weekly frequency. This has the advantages of providing a stable use of resources such as vessels and port berths, easing transshipments in ports, as it allows connecting time windows of communicating services, to be minimized, and provide customers with an easily understandable service catalogue of the next departure, and its transit time (as a local bus-route with a fixed frequency).

The number of vessels required to enforce weekly frequency on a service, can be bounded by:

$$d_s/s_*^v + 24 \cdot n_s \leq \tau_s \cdot 24 \cdot 7 \quad (5.1)$$

where d_s is the roundtrip distance of s , and n_s is the number of port calls on the service. Using the same assumptions, the weekly running costs c_s of a service s can be calculated as the sum of the daily time charter rate T^v per vessel, the idling bunker consumption per day f_0^v and the sailing bunker consumption $d_s/s_*^v/24 \cdot f_*^v$

$$c_s = 7 \cdot \tau_s \cdot T^v + (7 \cdot f_0^v + d_s/s_*^v/24 \cdot f_*^v) \cdot b \quad (5.2)$$

where b is the cost of bunkers in \$ per metric tonne. Note that the unit of d_s is *nautical miles per week* ($\frac{nm}{week}$) as the whole distance is traversed each week, as the service has a weekly frequency. Following we get the units $\frac{nm}{week} / \frac{nm}{hour} / \frac{hour}{day} \cdot \frac{mt}{day} = \frac{mt}{week}$, for the last term, $d_s/s_*^v/24 \cdot f_*^v$.

Vessels are assumed to sail at design speed s_*^v on all legs. The distance of the service might not exactly fit this distance, as seen in equation 5.1 where the number of vessels is rounded up. This corresponds to the vessels lying still for a while. In practice a vessel will be able to use this time to sail a bit slower than the design speed, and due to the polynomial relation between speed and bunker consumption, it could save bunker cost. Hence the calculated bunker costs are upper bounds on actual bunker costs.

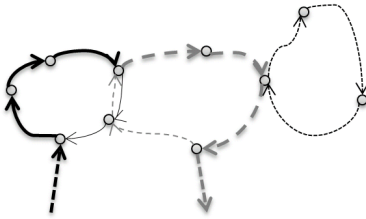
Given a graph $G(P, A)$, a set of demands K , a set of vessel classes V and data on above mentioned cost, volume, capacity, etc. the problem is to find a maximal profit liner shipping network.

5.3.1 Service flow

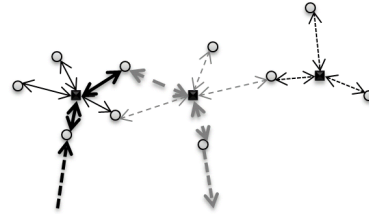
Current mathematical models of liner shipping network design problems use arc flow formulations, an example is seen in Figure 5.3(a). These have both flow variables $f_{i,j}$ from port i to j and network decision variables, $\delta_{i,j}$ of some form, related to the arcs of the graph. This works well when considering simple cycles. But in real world services it is common practice to have butterfly calls, for reasons mentioned in the introduction.

The issue arising with arc flow formulations, which effect butterfly services, can be illustrated by Figure 5.1(b). A node which is called twice, A , will have flow entering from two directions C and X , and flow leaving in two directions B and X . The flow entering from one of these, can only leave on one of the two leaving arcs. E.g. a demand entering from B should exit towards X , but if the formulation does not explicitly handle this, it can equally well exit towards A . The problem in this (flow entering from B leaving towards A) can be perceived in two ways; The flow cheating on capacity on the legs $A \rightarrow X \rightarrow A$ or cheating on paying transshipment costs in A .

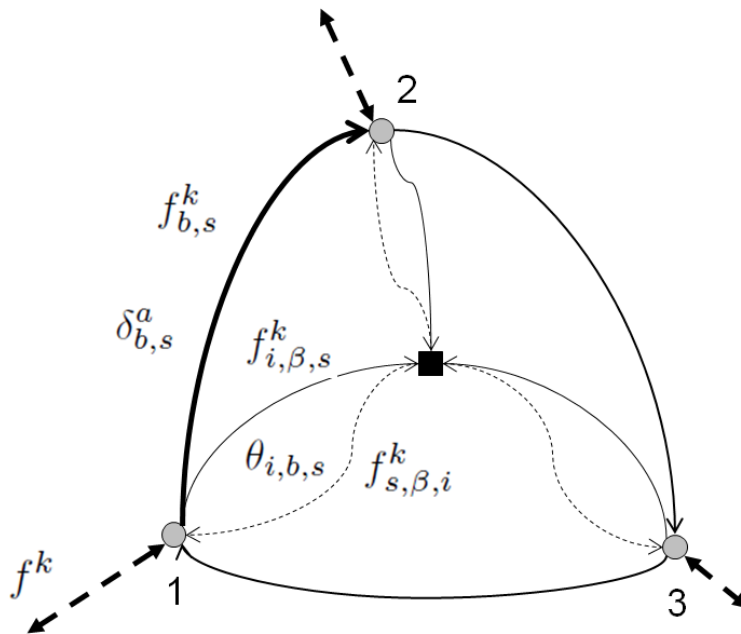
This issue can partly remedied as done in Reinhardt and Pisinger [11], which allows semi simple cycles, i.e. a vessel rotation can have a single butterfly port.



(a) Three services in an arc flow formulation. A flow path entering the first service, transshipping for the second and unloading at the flow destination is shown.



(b) The same three services in an service flow formulation. The service nodes are added in the middle of the three services, which then are connected with their called ports by arcs, opened by binary variable $\theta_{i,b,s}$ and flow by continuous variable $f_{i,b,s}^k$. A flow path entering the first service, transshipping for the second and unloading at the flow destination is shown.



(c) A three port service shown both in an arc and service formulation. A flow path loading in port 1 and unloading in port 2 is shown. Leg flow variables $f_{i,\beta,s}^k$ and $f_{s,\beta,i}^k$ are indicated next to the port to service arcs from port 1 to the service node. Binary variable for opening the arcs $\theta_{i,b,s}$ is shown in between the leg flow arcs. Port to port flow variable $f_{b,s}^k$ is shown next to the arc between port 1 and port 2, where the auxiliary port to port variable $\delta_{b,s}^{a(i,j)}$ indicating whether this arc is open, is also shown.

Figure 5.3: Edge flow modelling 5.3(a) as opposed to service flow modeling 5.3(b). Both are shown in 5.3(c)

The motivation for a service flow formulation comes from the observation that flow, for instance from a feeder port to a hub port, in a arc flow formulation is seen as a path between a number of feeder ports, finally arriving at the hub port. Rather, one could consider it by connecting the feeder port with a service, which then again was connected to the hub port. An example of this

is seen in figure 5.3(b), the square black nodes represent services, the round grey nodes represent ports, arcs pass from port-nodes to service-nodes only.

Flow variables $f_{i,b,s}^k$ represent flow of some demand k from port i to the b 'th call of service s , $f_{s,b,i}^k$ represent flow from the b 'th call of s to i . The auxiliary measured flow variables $f_{b,s}^k$ are the flow of demand k on the b 'th leg of service s . Binary decision variables $\theta_{i,b,s}$ are then set to 1 iff i is the b 'th port on service s .

This formulation allows any number and sequence of recurring calls to a butterfly port. Some problems arise from this formulation that must be tackled in different ways.

Number of port calls per service Since B and S are indices of flow and decision variables, these are input parameters of the models. B can be seen as an upper bound on number of port calls on a service. By using binary variables Ψ_{b_s} in the model, a service can be of any length less than or equal to B .

Weekly frequency Large ocean carriers predominantly use weekly frequency on their services. This is enforced in the generated services, by the equation (5.1), which ensures that the service can traverse the distance d_s with a sailing speed at or less than s_s^v . Thus the cost of the service will be a lower bound of the real cost as the vessels in practise may sail slower, allowing for less fuel spent due to the quadratic fuel consumption curve (see Brouer et al. [5] for details).

Vessel-Port Draught compatibility If vessels have larger draught than some port allows, they are not able to call it, hence the decision variable representing this must be fixed at 0.

Available vessels per vessel class Each vessel class v , has a limited number z_v of vessels available, all services using the same vessel class must adhere to this.

Capacity The capacity of each leg of the service must be respected. Looking at figure 5.3(c), it can be seen that the flow on leg 1 (from port 1 to port 2) of demand k , $f_{1,s}^k$, must adhere to:

$$f_{1,s}^k \geq \sum_i (f_{i,1,s}^k - f_{s,1,i}^k) \quad \text{and} \quad f_{1,s}^k \geq \sum_i (f_{i,1,s}^k - f_{s,1,i}^k + f_{i,3,s}^k - f_{s,3,i}^k) \quad (5.3)$$

as for all demands k and services s , at least one leg b , will have no flow $f_{b,s}^k = 0$, otherwise the flow-path would be taking the full cycle, which can not be optimal. To arrive at the flow on leg b we to need consider the largest of flow deltas ($f_{i,\beta,s}^k - f_{s,\beta,i}^k$) sums, for $\beta \in \text{circ}(b, \dots, B-1)$ previous legs. This can be generalized to:

$$\sum_{\text{circ}(b > \beta \geq \alpha)} \sum_{p \in P} (f_{i,\beta,s}^k - f_{s,\beta,i}^k) \leq f_{b,s}^k \quad \forall s, k, b, \alpha \in B \setminus \{b\} \quad (5.4)$$

where the first summation implies: for each previous call α we need to sum for legs β until the considered call b . The call b can then either be before α as seen from the 0 'th leg, giving: ($b \leq \alpha$) where we sum over β legs: $\beta < b$ and $\beta \geq \alpha$. Or b can be after α as seen from the 0 'th leg, giving: ($b > \alpha$) were we sum β legs: $b > \beta \geq \alpha$. The capacity can then be imposed on variables $f_{b,s}^k$.

The auxiliary binary decision variable $\delta_{b,s}^a$ is set to 1 iff arc a is the b 'th arc used by service s . If the draft of a port i does not allow call by a vessel v : $\text{draft}(i) < \text{draft}(v(s))$, the corresponding variables $\theta_{i,b,s}$, $\delta_{b_s}^{a(i,j)}$ and $\delta_{b_s}^{a(j,i)}$ are set to 0. The non-negative integer variable τ_s is the number of vessels used on service s .

5.3.2 Service flow formulation of the LSN DP

The LSN DP can then be formulated as the following mixed integer programming model:

$$\begin{aligned} \max \quad & \sum_{k \in K} q^k f^k - \sum_{k \in K} \tilde{q}^k (F_k - f^k) - \sum_{i \in P} u_i \sum_{k \in K} \sum_{s \in S} \sum_{b \in B} (f_{i,b,s}^k + f_{s,b,i}^k) \\ & - \sum_{s \in S} \sum_{i \in P} \sum_{b \in B} w_{i,v(s)} \theta_{i,b,s} - \sum_{s \in S} c_s \end{aligned} \quad (5.5)$$

$$\text{s.t.} \quad \sum_{i \in P} \sum_{b \in B} f_{i,b,s}^k = \sum_{i \in P} \sum_{b \in B} f_{s,b,i}^k \quad \forall k \in K, \forall s \in S \quad (5.6)$$

$$\sum_{s \in S} \sum_{b \in B} f_{d(k),b,s}^k = f^k \quad \forall k \in K \quad (5.7)$$

$$\sum_{s \in S} \sum_{b \in B} f_{i,b,s}^k = \sum_{s \in S} \sum_{b \in B} f_{s,b,i}^k \quad \forall k \in K, \forall i \in P \setminus \{o(k), d(k)\} \quad (5.8)$$

$$\sum_{\text{circ}(b > \beta \geq \alpha)} \sum_{i \in P} (f_{i,\beta,s}^k - f_{s,\beta,i}^k) \leq f_{b,s}^k \quad \forall s \in S, \forall k \in K, \quad (5.9)$$

$$\sum_{k \in K} f_{b,s}^k \leq c_{v(s)} \sum_{i \in P} \theta_{i,b,s} \quad \forall b \in B, \forall s \in S \quad (5.10)$$

$$\sum_{k \in K} (f_{i,b,s}^k + f_{s,b,i}^k) \leq 2c_{v(s)} \theta_{i,b,s} \quad \forall i \in P, \forall b \in B, \forall s \in S \quad (5.11)$$

$$f_{i,b,s}^k + f_{s,b,i}^k \leq F_k \theta_{i,b,s} \quad \forall i \in P, \forall b \in B, \forall s \in S, \forall k \in K \quad (5.12)$$

$$\theta_{i,b,s} + \theta_{j,b+1,s} \leq 1 + \delta_{b,s}^{a(i,j)} \quad \forall i, j \in P, \forall b \in B, \forall s \in S \quad (5.13)$$

$$2 * \delta_{b,s}^{a(i,j)} \leq \theta_{i,b,s} + \theta_{j,b+1,s} + \Psi_{b+1,s} \quad \forall i, j \in P, \forall b \in B, \forall s \in S \quad (5.14)$$

$$\theta_{i,b,s} + \Psi_{b+1,s} + \theta_{j,1,s} \leq 2 + \delta_{b,s}^{a(i,j)} \quad \forall i, j \in P, \forall b \in B, \forall s \in S \quad (5.15)$$

$$\sum_{i \in P} \theta_{i,b,s} + \Psi_{b,s} = 1 \quad \forall b \in B, \forall s \in S \quad (5.16)$$

$$\sum_{a \in A} \delta_{b,s}^a + \Psi_{b,s} = 1 \quad \forall b \in B, \forall s \in S \quad (5.17)$$

$$\Psi_{b,s} \leq \Psi_{b+1,s} \quad \forall b \in B \setminus \{|B|\}, \forall s \in S \quad (5.18)$$

$$\sum_{a \in A} \sum_{b \in B} d_a / s_*^v \cdot \delta_{b,s}^a + 24 \sum_{i \in P} \sum_{b \in B} \theta_{i,b,s} \leq 24 \cdot 7 \cdot \tau_s \quad \forall s \in S \quad (5.19)$$

$$\sum_{s \in v} \tau_s \leq z_v \quad \forall v \in V \quad (5.20)$$

$$0 \leq f^k, f_{i,b,s}^k, f_{s,b,i}^k, f_{b,s}^k \leq F_k \quad \forall k \in K, \forall i \in P, \forall b \in B, \forall s \in S \quad (5.21)$$

$$\theta_{i,b,s}, \delta_{b,s}^a \in \{0, 1\} \quad \forall i \in P, a \in A, \forall b \in B, \forall s \in S \quad (5.22)$$

$$\Psi_{b,s}, \tau_s \in \mathbb{Z}_+ \quad \forall b \in B, \forall s \in S \quad (5.23)$$

The objective is to maximize the profit of flowed cargo on the opened network, by optimizing the revenue subtracted a penalty for cargo not flowed, done in the two first terms of (5.5), the container move cost are subtracted in the third term, as are the port call cost in the fourth term and lastly the costs of deployed vessels and bunker cost in c_s .

The flow must adhere to flow conservation in service nodes ensured by constraint (5.6) and port nodes by constraint (5.8). Constraints (5.7) allows flow to enter at the demands source node. The $f_{b,s}^k$ variables must be set by constraint (5.9). The capacity is enforced in constraint (5.10). We

can only flow on open arcs, this is handled by constraint (5.11), using service capacity as big M, which is also tightened by constraints (5.12). Constraints (5.13) - (5.18) ensure proper construction of services, i.e. Arc decision variables $\delta_{b,s}^a$ are set in (5.13). Constraints (5.14) and (5.15) links variables $\theta_{i,b,s}$ and $\delta_{b,s}^a$. Each port call must have exactly one call as handled by constraints (5.16) and (5.17). Constraint (5.18) propagates the variable $\Psi_{b,s}$ through b if set. Weekly frequency is enforced as lower bound on service distance with constraint (5.19). The number of vessels is constrained by (5.20). Variable bounds are given in (5.21) - (5.23).

To further tighten the formulation a number of additional constraints can be added:

$$\sum_{j \in P} \sum_{b \in B} \delta_{b,s}^{a(i,j)} = \sum_{j \in P} \sum_{b \in B} \delta_{b,s}^{a(j,i)} \quad \forall i \in P, \forall s \in S \quad (5.24)$$

$$\sum_{i \in P} \sum_{j \in P} \sum_{b \in B} \delta_{b,s}^{a(i,j)} = \sum_{i \in P} \sum_{b \in B} \theta_{i,b,s} \quad \forall s \in S \quad (5.25)$$

$$\theta_{i,b,s} + \theta_{i,b+1,s} \leq 1 \quad \forall i \in P, \forall b \in B, \forall s \in S \quad (5.26)$$

$$\sum_{k \in K} d_{o(k),d(k)} f^k \leq \sum_{s \in S} \sum_{a \in A} \sum_{b \in B} c_{v(s)} d_a \delta_{b,s}^a \quad (5.27)$$

$$w_{i,v(s)} \theta_{i,b,s} \leq \sum_{k \in K} (f_{i,b,s}^k + f_{s,b,i}^k) (q^k + \bar{q}^k - u_{o(k)} - u_{d(k)}) \quad \forall i \in P, \forall b \in B, \forall s \in S \quad (5.28)$$

where constraints (5.24) - (5.26) tighten the domain of the binary variables. Constraint (5.24) gives that service s for a port i must enter and leave an equal number of times, with regards to the auxiliary $\delta_{b,s}^{a(i,j)}$ variables. Constraint (5.25) gives that the sum of port calls given by $\delta_{b,s}^a$ and $\theta_{i,b,s}$ variables must equal. Constraint (5.26) ensures that two consecutive port calls can not call the same port, i . Constraint (5.27) has the total flowed volume multiplied by its direct distance as a lower bound for all services roundtrip distances multiplied by their capacity, as an optimal capacity allocation would satisfy this. It is required by constraint (5.28) that revenue and goodwill of demand loaded and unloaded in a port call, subtracted the first load and last discharge move, exceeds the port call cost.

5.3.3 Symmetry

The formulation will suffer considerably of degenerate optimal solutions, due to the sets B and S , some of the degeneracy can be dealt with by two types of symmetry breaking constraints.

$$\sum_{k \in K} (f_{i,b,s}^k + f_{s,b,i}^k) \leq \sum_{k \in K} (f_{i,1,s}^k + f_{s,1,i}^k) \quad \forall b \in B \setminus \{1\}, \forall s \in S \quad (5.29)$$

$$\sum_{a \in A} \sum_{b \in B} \delta_{b,s}^a \cdot d_a \leq \sum_{a \in A} \sum_{b \in B} \delta_{b,s+1}^a \cdot d_a \quad \forall s \in v, \forall v \in V, \quad (5.30)$$

The first considers the flow on a service, as the legs of the service are numbered from $\{1, \dots, |B|\}$, an identical service can be generated by shifting the numbering of all legs by one. Constraint (5.29) will break this symmetry by requiring that the 1'th call of the service always has the most load and unload moves. Note that if several calls have the same number of moves, symmetry will still exist. Constraint (5.30) considers services of the same vessel class, requiring these to be ordered by non-decreasing distance, this will alleviate symmetry arising in the formulation.

5.3.4 Bounding of $|S|$ and $|B|$

The solution space of the model will be bounded by the size of the sets S and B , as this will impose artificial limits on how many and long the generated services can be. Hence it is valuable to know upper bounds that these can take in a non constrained version, to be able to set S and

Instance	Scenario	#Ports	#Demands	# Feeder 450	# Feeder 800
Baltic	Low	12	22	3	2
	Base	12	22	4	2
	High	12	22	5	2
WAF	Low	19	38	11	22
	Base	19	38	14	28
	High	19	38	17	34

Table 5.1: The **Baltic** and **WAF** instances from LINER-LIB 2012. # Ports is the number of ports, # Demands is the number of demands, # Feeder 450 and # Feeder 800 is the number of vessels available of the vessel classes.

B accordingly. A lower bound can only be $0 \leq |S|$ and $0 \leq |B|$ as the 0 solution is feasible and will in some instances be optimal. For an upper bound, consider a solution where each demand k had its own dedicated direct service, thus using two legs. This gives the bounds $|S||B| \leq 2|K|$, $|S| \leq |K|$, $|B| \leq 2|K|$.

5.3.5 Problem Complexity

The problem is NP-hard by reduction from the TSP. Given a set of nodes $i, j \in P$ with distances d_{ij} between the nodes the TSP problem asks to find the shortest Hamiltonian cycle visiting all nodes P . For a given instance of TSP we can reduce it to an instance of LSNDP by having only one vessel class v , and only one vessel $z_v = 1$. Let the demand f^k between each pair of nodes P be 1 and the penalty \tilde{q}^k for not shipping the cargo be infinitely large. The sailing bunker consumption between ports i and j is set to d_{ij} and the idling bunker consumption is set to zero. The sailing time between all pair of nodes is set to $24 \cdot 7/|P|$, such that any route visiting all nodes takes exactly one week. Finally, the transshipment move cost is set to infinity and the port call cost is set to 0.

5.4 Computational results

The Mixed Integer Program of (5.5) - (5.28), SFM has been implemented as an MIP-model and solved by CPLEX 12.2 on a Intel i5 @ 2.53 GHZ with 3 GB of RAM. It has been run with the LINER-LIB 2012 instances and compared with the best solutions found in Brouer et al. [5].

Instances The LINER-LIB 2012 benchmark is a set of publicly available instances for the LSNDP presented in Brouer et al. [5]. All relevant data for a network design instance is available, distances between ports d_a ; Vessel classes v described by their capacity, drafts, cost and speed limitations; a list of ports i described by their draft limits, call cost and move cost. Seven instances of varying size is available, each with a fleet of vessels available. Each instance has a demand list K with a volume and revenue that can be flowed. As the size of the SFM is proportional to the size of the number of port calls per service B and the number of services S , this imposes an additional constraint on the model. For the run instances B and S have been set high, to limit the constraining impact, but due to memory shortage for large B and S values a balance has been struck.

SFM results The model has been run on the two smallest instances of LINER-LIB 2012, **Baltic** and **WAF**, Table 5.1 has details on the sizes of these. Both of the instances have one main port (Bremerhaven and Algeiras) and all demands are to or from this main port to feeder ports. Revenue and all costs parameters outlined in the objective Equation (5.5) is included: move cost for loading and unloading containers; port call cost, vessel and bunker costs (both propulsion and idling bunker consumption).

Instance	Scenario	\tilde{q}^k	Frequency	Objective	U.B.	Time	$\sum_s B $	$ S $	# vessels
Baltic	Low	0	7	427,485	611,015	3,600	9/20	2/2	4/5
	Base	0	7	408,771	669,774	3,600	10/20	1/2	4/6
	High	0	7	636,152	657,021	3,600	10/20	1/2	4/7
WAF	Low	0	7	1,940,817	6,051,697	10,800	10/24	3/4	11/33
	Base	0	7	3,372,618	6,399,041	10,800	24/24	4/4	26/42
	High	0	7	3,899,767	6,614,613	10,800	24/24	4/4	28/51

Table 5.2: Test results for the **Baltic** and **WAF** instances run by SFM. \tilde{q}^k is the used goodwill penalty for rejected cargo. The *Frequency* of the generated services. *Objective* is the objective value of the best found solution by SFM, *U.B.* is the upper bound as given by CPLEX for the SFM formulation, *Time* is the time limit in seconds. $\sum_s |B|$ is the total number of port calls used of the allowed. $|S|$ is the number of services used of the allowed. # vessels is the number of vessels used of the allowed.

Instance	Scenario	Z	Q	c_v	c_b	c_p	c_m	c_t	F
Baltic	Low	427,485	2,389,910	-225,400	-226,833	-273,488	-1,236,703	0	0
	Base	408,771	2,020,328	-140,000	-243,008	-272,637	-954,365	-1,546	0
	High	636,152	2,641,046	-112,000	-225,343	-277,309	-1,390,241	0	0
WAF	Low	1,940,817	4,508,620	-597,800	-833,453	-124,516	-1,012,034	0	0
	Base	3,372,618	9,536,510	-1,162,000	-2,210,227	-322,030	-2,241,530	-228,105	0
	High	3,899,767	9,907,400	-968,800	-2,311,786	-327,024	-2,400,023	0	0

Table 5.3: Revenue and cost details for the best solution for **Baltic** and **WAF** instances found with SFM. Z is the objective value, Q is the revenue of flowed cargo, c_v is vessel cost, c_b is bunker cost, c_p is port call cost, c_m is local move cost, c_t is transshipment move cost and Y the sum of goodwill penalties for rejected cargoes.

The instances have been run with zero goodwill penalty $\tilde{q}^k = 0$ for demand not flowed. This resembles a scenario where a carrier solely focuses on optimizing profits. The results are given in Tables 5.2 and 5.3. None of the instances can be solved to optimality, although one has a gap of just 3 %. Profitable solutions are found for all instances. Due to their size the **WAF** instances are considerably harder than the **Baltic** instances. No instance use all vessels, but it cannot be ruled out that B and S are restrictive and thus limits the use of vessels. Table 5.4 shows the considerable size of the MIPs. Referring to Table 5.3 it can be seen that only two instances use transshipment, and one of these on a very small scale. This is reasonable as all demand is from or to a main port. An example of a solution can be seen in Table 5.8.

Benchmark Instance Algorithm In the spirit of the LINER-LIB 2012, the results of the SFM model have been compared with the results from Brouer et al. [5], the Benchmark Instance Algorithm (BIA). The SFM model has some differences from the model and algorithm of BIA, so these are not directly comparable, but by highlighting the differences it is still meaningful to discuss.

BIA allows biweekly services in some cases. The average frequencies of the best BIA result services, can be seen in Table 5.5, where the average frequency is closer to 14 than 7, for all instances. This allows solutions serving low demand ports cheaper, as larger (and thus generally with cheaper unit cost) vessels can serve these less frequently. To make the results of SFM closer

Instance	# Rows	# Cols	# Non Zero's
Baltic	19,949	14,201	621,018
WAF	54,943	46,654	1,001,869

Table 5.4: MIP Instance sizes for **Baltic** and **WAF** with SFM. #Rows is the number of rows. #Cols is the number of columns. #NonZero's is the number of non zero matrix entrances.

Instance	Scenario	$ S $	$\sum_s B_s $	Avg. speed	Avg. frequency	Best Objective
Baltic	Low	5	27	13.6	13.6	235,044
	Base	4	29	13.3	12.6	325,305
	High	5	32	12.9	13	609,700
WAF	Low	7	63	12.3	12.2	4,486,261
	Base	11	68	11.6	11.7	5,565,389
	High	12	68	11.4	11.7	6,220,044

Table 5.5: Details of the best results found by the BIA algorithm. $|S|$ is the number of services used and $\sum_s |B_s|$ is the total number of port calls in the solution. Avg. speed is the averaged speed of the services, weighted by their number of vessels. Avg. frequency is the averaged frequency of the services, weighted by their number of vessels.

to the BIA results, SFM has been modified to allow biweekly services (by halving port call, bunker and vessel cost, halving capacity and doubling the right hand side of Equation (5.19), and the goodwill penalty has been set to $\tilde{q}^k = 1000$, as done in BIA. The results of the modified SFM can be seen in Tables 5.6 and 5.7. Comparing these against the original runs (Tables 5.2 and 5.3), the consequences of halving the vessels capacity can be seen, all port calls B and services S are used in all instances. This is likely one of the reason for the worse solutions. Another is the goodwill penalty \tilde{q}^k , which lowers the overall objective, but also seems to make the problem significantly harder, with much worse gaps. A reason for this is Equation (5.28) being looser with increased goodwill penalty. Details of the best solutions are found in Table 5.3 and 5.7 showing the distribution between revenue and the different cost components.

Still there are other differences between SFM and BIA which can not be mitigated. BIA does not enforce strict (bi)weekly frequency, but allows slight deviations. This gives up to 9 % extra capacity. BIA optimizes on the speeds of the generated services, allowing it to slow steam, where beneficial and thus get lower bunker costs. Or allows the services to speed up to get additional cargo where possible, details on the sailing speeds can be seen in Table 5.5. Lastly BIA is a randomized algorithm, which gives different solutions depending on the seed. We are comparing against the best results from 10 randomized runs. The results of BIA is reported in Brouer et al. [5] over 180 days, and are minimizing the cost subtracted the revenue. The SFM is run on weekly values and is maximizing the revenue subtracted the cost. Hence all values taken from BIA results have been multiplied by $-180/7$ to be comparable with the results of SFM.

On the other hand SFM allows any number of butterfly ports, opening the solution space, where BIA only allows one butterfly port with two port calls. SFM is restricted in its solution space if the sizes of $|B|$ and $|S|$ are limiting. Due to lack of memory the model has not been tested on larger instances.

Comparing results Looking at Tables 5.5 and 5.6 it can be seen that BIA finds a better solution in all cases, but except for one instance the BIA solutions are within the upper bound of SFM. One reason is that the sizes of $|B|$ and $|S|$ in SFM is severely restricting, as BIA uses significantly more port calls and services. The upper bounds of SFM exceeds the best found solutions of BIA except for the instance High WAF instance, which must be due to the differences in the two formulations. In all cases the distance between the BIA solution and the SFM upper bound is small, thus validating the quality of the BIA solutions as giving very good results close to a comparable upper bound.

5.5 Conclusion

A novel model has been presented for the LSNDP. To the best of our knowledge the model is the first method to fully allow any number of butterfly ports, allowing with more than two port calls, a crucial property as many real world services often call a number of ports more than once on the

Instance	Scenario	\tilde{q}^k	Frequency	Objective	U.B.	Time	$\sum_s B $	$ S $	# vessels
Baltic	Low	1,000	14	-265,117	794,524	3,600	20/20	2/2	5/5
	Base	1,000	14	-134,687	830,563	3,600	20/20	2/2	5/6
	High	1,000	14	-183,348	849,487	3,600	20/20	2/2	6/7
WAF	Low	1,000	14	411,317	5,650,061	10,800	24/24	4/4	20/33
	Base	1,000	14	1,059,352	5,665,930	10,800	24/24	4/4	20/42
	High	1,000	14	1,281,583	5,825,944	10,800	24/24	4/4	20/51

Table 5.6: Test results for the **Baltic** and **WAF** instances run by SFM. \tilde{q}^k is the used goodwill penalty for rejected cargo. The *Frequency* of the generated services. *Objective* is the objective value of the best found solution by SFM, *U.B.* is the upper bound as given by CPLEX for the SFM formulation, *Time* is the time limit in seconds. $\sum_s |B|$ is the total number of port calls used of the allowed. $|S|$ is the number of services used of the allowed. # vessels is the number of vessels used of the allowed.

Instance	Scenario	Z	Q	c_v	c_b	c_p	c_m	c_t	F
Baltic	Low	-265,117	3,417,398	-151,900	-298,605	-425,925	-1,841,615	-18	-964,453
	Base	-134,687	3,455,032	-108,500	-284,312	-425,921	-1,870,986	0	-900,000
	High	-183,348	3,439,210	-100,800	-323,047	-489,550	-1,911,161	0	-798,000
WAF	Low	411,317	8,491,320	-622,300	-1,508,075	-197,533	-2,100,095	0	-3,652,000
	Base	1,059,352	8,828,980	-444,500	-1,431,523	-203,320	-2,135,286	0	-3,555,000
	High	1,281,583	9,020,610	-355,600	-1,478,053	-226,845	-2,121,531	0	-3,557,000

Table 5.7: Revenue and cost details for the best solution for **Baltic** and **WAF** instances found with SFM. Z is the objective value, Q is the revenue of flowed cargo, c_v is vessel cost, c_b is bunker cost, c_p is port call cost, c_m is local move cost, c_t is transshipment move cost and Y the sum of goodwill penalties for rejected cargoes.

Service	Vessel Class	Num Vessels	Rotation
Service 1	450	9	<i>Algeciras</i> → <i>Douala</i> → <i>Algeciras</i> → <i>Tema</i> → <i>Algeciras</i> → <i>Conakry</i>
Service 2	450	8	<i>Algeciras</i> → <i>Lome</i> → <i>Algeciras</i> → <i>Luanda</i> → <i>Tema</i> → <i>Dakar</i>
Service 3	800	5	<i>Dakar</i> → <i>Algeciras</i> → <i>Dakar</i> → <i>Algeciras</i> → <i>Dakar</i> → <i>Apapa</i>
Service 4	800	6	<i>Dakar</i> → <i>Algeciras</i> → <i>Cotonou</i> → <i>Dakar</i> → <i>Algeciras</i> → <i>Abidjan</i>

Table 5.8: The best found solution for the **WAF** High case with seven days frequency and no goodwill penalty. Note that Service 3 has both Dakar and Algeciras as butterfly port, Dakar with three port calls.

rotation. The model has been run on two of the benchmark instance of LINER-LIB 2012 providing solutions for these.

Due to the large number of variables and constraints used by the model it is unable to solve the instances to full optimality. Future work could address this in a number of ways, a MIP based heuristic could be developed to find good solutions fast. Decompositions of the model could be investigated, perhaps exploiting the separate services and flow structure to deal with these problems separately, or thirdly, tighter constraints and cuts could be devised for the problem.

Acknowledgements

The authors gratefully acknowledge the support from The Danish Council for Strategic Research to the ENERPLAN project. The authors would like to thank Berit Dangaard Brouer and Christian Vad Karsten for valuable comments. We thank two reviewers for very qualified and constructive comments, this work has been greatly enhanced due to them.

Bibliography

- [1] Time charter rates according to cbs, statistics netherlands. <http://www.cbs.nl/en-GB/menu/themas/dossiers/globalisering/publicaties/artikelen/archief/2009/2009-2801-wm.htm>, 2009.
- [2] Bunker prices from bunkerworld. <http://www.bunkerworld.com/prices/index/bw380>, 2012.
- [3] Richa Agarwal and Ozlem Ergun. Ship scheduling and network design for cargo routing in liner shipping. *Transportation Science*, 42(2):175–196, 2008.
- [4] J. Fernando Alvarez. Joint routing and deployment of a fleet of container vessels. *Maritime Economics and Logistics*, 11:186–208, 2009.
- [5] Berit D. Brouer, J. Fernando Alvarez, Christian E. M. Plum, David Pisinger, and Mikkel M. Sigurd. A base integer programming model and benchmark suite for liner shipping network design. *Transportation Science*, 2013. Accepted by Transportation Science.
- [6] Marielle Christiansen, Kjetil Fagerholt, and David Ronen. Ship routing and scheduling: status and perspectives. *Transportation Science*, 38(1):1–18, 2004.
- [7] Marielle Christiansen, David Ronen, Bo Nygreen, and Kjetil Fagerholt. Maritime transportation. In *Handbooks in Operations Research and Management Sciences*, pages 189–284. North Holland, 14 edition, 2007.
- [8] Shahin Gelareh and David Pisinger. Fleet deployment, network design and hub location of liner shipping companies. *Transportation Research Part E: Logistics and Transportation Review*, 47:947 – 964, 2011.
- [9] Qiang Meng and Shuaian Wang. Liner shipping service network design with empty container repositioning. *Transportation Research Part E: Logistics and Transportation Review*, 47(5): 695–708, 2011.
- [10] Qiang Meng, Shuaian Wang, Henrik Andersson, and Kristian Thun. Containership routing and scheduling in liner shipping: Overview and future research directions. *Transportation Science*, 2013. To appear in.
- [11] Line B. Reinhardt and David Pisinger. A branch and cut algorithm for the container shipping network design problem. *Flexible Services And Manufacturing Journal*, 2011.
- [12] Koichi Shintani, Akio Imai, Etsuko Nishimura, and Stratos Papadimitriou. The container shipping network design problem with empty container repositioning. *Transportation Research Part E: Logistics and Transportation Review*, 43(1):39–59, 2007.
- [13] Martin Stopford. *Maritime Economics*. Routledge, third edition, 2009.
- [14] Shuaian Wang and Qiang Meng. Schedule design and container routing in liner shipping. *Transportation Research Record: Journal of the Transportation Research Board*, 2222:25–33, 2011.
- [15] Shuaian Wang and Qiang Meng. Robust schedule design for liner shipping services. *Transportation Research Part E: Logistics and Transportation Review*, 48(6):1093 – 1106, 2012.