A Node Flow Model for the Inflexible Visitation Liner Shipping Fleet Repositioning Problem with Cargo Flows

Kevin Tierney and Rune Møller Jensen

IT University of Copenhagen, Copenhagen, Denmark
{kevt,rmj}@itu.dk

Abstract. We introduce a novel, node flow based mathematical model for the fixed-time version of a central problem in the liner shipping industry called the Liner Shipping Fleet Repositioning Problem (LSFRP). We call this version of the problem the Inflexible Visitation LSFRP (IVLSFRP). During repositioning, vessels are moved between routes in a liner shipping network. Shipping lines wish to reposition vessels as cheaply as possible without disrupting the cargo flows of the network. The LSFRP is characterized by chains of interacting activities with a multi-commodity flow over paths defined by the activities chosen. We introduce two versions of a node flow based model that exploit the fixed activity times of the IVLSFRP's graph to handle cargo demands on the nodes of the graph, instead of the arcs, significantly reducing the number of variables. Using this model in CPLEX, we are able to solve 12 previously unsolved IVLSFRP instances to optimality. Additionally, we improve the solution time on every instance in the IVLSFRP dataset, sometimes by several orders of magnitude.

1 Introduction

Liner shipping networks are the lifeblood of the world economy, providing cheap and reliable freight services between nations around the globe. Vessels are regularly repositioned between services in liner shipping networks to adjust networks to the continually changing world economy. Repositioning vessels involves creating a plan for a set of vessels out of a number of cost saving (or revenue earning) activities that moves (*repositions*) the vessels to a particular route in the network. Since repositioning a single vessel can cost hundreds of thousands of US dollars, optimizing the repositioning activities of vessels is an important problem for the liner shipping industry.

The Liner Shipping Fleet Repositioning Problem (LSFRP) with cargo flows, first introduced in [18], consists of finding sequences of activities that move vessels between services in a liner shipping network while respecting the cargo flows of the network. The LSFRP maximizes the profit earned on the subset of the network affected by the repositioning, balancing sailing costs and port fees against cargo and equipment revenues, while respecting important liner shipping specific constraints dictating the creation of services and movement of cargo. The Inflexible Visitation LSFRP (IVLSFRP) is a simplified version of the LSFRP in which the time of every visitation (i.e. port call) a vessel may undertake is known in advance. Such visitations are called *inflexible*. In the full LSFRP, the vessel entry/exit time of some visitations is unknown, requiring solution methods to schedule these visitations if they are chosen for a repositioning.

The number of variables in the arc flow model from [16,18] grows linearly in the number of graph arcs multiplied by the number of cargo demands, and, as a result, it quickly runs out of memory or CPU time on large instances. In this paper, we introduce a novel node flow model for the IVLSFRP that fits into memory even on large instances, thanks to the fact that the number of nodes tends to be significantly lower than the number of arcs. The node flow model exploits the fact that when all visitations are inflexible, the sequencing of demands is known in advance. We use this fact in order to model demands based on the graph nodes they can pass through. We provide two versions of the node flow model that each handle equipment (i.e. empty container) flows in different ways; one version uses an arc flow formulation, and the other embeds equipment into the problem's demand structure.

The node flow model solves 12 previously unsolved IVLSFRP instances to optimality within 5 hours. The node flow model solves IVLSFRP instances to optimality significantly faster than the arc flow approach, and is even able to solve a number of instances in under a second that require 15 minutes or more with the arc flow approach.

This paper is organized as follows. We first give an overview of the IVLSFRP in Section 2, followed by the arc flow model presented in [16] in Section 3. We then introduce our node based formulation of the IVLSFRP in Section 4. We provide a computational evaluation of the new model with IBM CPLEX 12.4 in Section 5 showing the improved performance of the node flow approach. Finally, we conclude in Section 6.

2 Liner Shipping Fleet Repositioning

We briefly describe the IVLSFRP, and refer readers to [16] for a more detailed description, as well as a description of the full version of the LSFRP. Liner shipping networks consist of a set of cyclical routes, called services, that visit ports on a regular, usually weekly, schedule. Liner shipping networks are designed to serve customer's cargo demands, but over time the economy changes and liner shippers must adjust their networks in order to stay competitive. Liner shippers add, remove and modify existing services in their network in order to make changes to the network. Whenever a new service is created, or a service is expanded, vessels must be *repositioned* from their current service to the service being added or expanded.

Vessel repositioning is expensive due to the cost of fuel (in the region of hundreds of thousands of dollars) and the revenue lost due to cargo flow disruptions. Given that liner shippers around the world reposition hundreds of vessels per year, optimizing vessel movements can significantly reduce the economic and environmental burdens of containerized shipping, and allow shippers to better utilize repositioning vessels to transport cargo. The aim of the IVLSFRP is to maximize the profit earned when repositioning a number of vessels from their initial services to a service being added or expanded, called the goal service.

Liner shipping services are composed of multiple *slots*, each of which represents a cycle that is assigned to a particular vessel. Each slot is composed of a number of *visitations*, which can be thought of as port calls, i.e., a specific time when a vessel is scheduled to arrive at a port. A vessel that is assigned to a particular slot sequentially sails to each visitation in the slot.

Vessel sailing speeds can be adjusted throughout repositioning to balance cost savings with punctuality. The bunker fuel consumption of vessels increases cubically with the speed of the vessel. *Slow steaming*, in which vessels sail near or at their minimum speed, therefore, allows vessels to sail more cheaply between two ports than at higher speeds, albeit with a longer duration (see, e.g., [12]).

Phase-out & Phase-in The repositioning period for each vessel starts at a specific time when the vessel may cease normal operations, that is, it may stop sailing to its scheduled visitations and go somewhere else. Each vessel is assigned a time when it may begin its repositioning, called its *phase-out* time. After this time, the vessel may undertake a number of different activities to reach the goal service. In order to complete the repositioning, each vessel must join slot on the goal service before a time set by the repositioning coordinator, called the *phase in* time. After this time, normal operations on the goal service begin, and all scheduled visitations on the service are to be undertaken. In other words, the repositioning of each vessel and optimization of its activities takes place in the period between two fixed times, the vessel's earliest phase-out time and the latest phase-in time of all vessels.

Cargo and Equipment Revenue is earned through delivering *cargo* and *equipment* (typically empty containers). We use a detailed view of cargo flows. Cargo is represented as a set of port to port demands with a cargo type, a latest delivery time, an amount of TEU[1] available, and a revenue per TEU delivered. We subtract the cost of loading and unloading each TEU from the revenue to determine the profit per TEU of a particular cargo demand. In contrast to cargo, which is a multi-commodity flow where each demand is a commodity with an origin and destination port, equipment can be sent from any port where it is in surplus to any port where it is in demand. We consider cargo demands and equipment consisting of either *dry* or *reefer* (refrigerated) cargo. Vessels have a limited capacity, and are therefore assigned a maximum number of reefer containers and a maximum number of all types of containers.

Sail-on-service (SOS) Opportunities While repositioning, vessels may use certain services to cheaply sail between two parts of the network. These are called *SOS opportunities*. The two vessels involved in SOS opportunities are referred to as the *repositioning vessel*, which is the vessel under the control of a repositioning coordinator, and the *on-service vessel*, which is the vessel assigned to a slot on the service being offered as an SOS opportunity. Repositioning vessels can use SOS opportunities by replacing the on-service vessel and sailing in its place for a portion of the service. SOS opportunities save significant amounts of money on bunker fuel, since one vessel is sailing where there would have otherwise been two. Using an SOS can even sometimes earn money through the leasing of the on-service vessel. Using an SOS is subject to a number of constraints, which are described in full in [16].

Asia-CA3 Case Study Figure 1 shows a subset of a real repositioning scenario in which a vessel must be repositioned from its initial service (the phase-out service), the Chennai-Express, to the goal service (the phase-in service), the Intra-WCSA. The Asia-CA3 service is offered as a SOS opportunity to the vessel repositioning from Chennai

[1] TEU stands for *twenty-foot equivalent unit* and represents a single twenty-foot container.

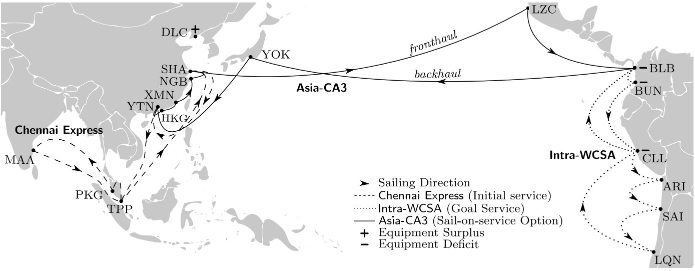


Fig. 1: A subset of a case study of the LSFRP, from [17].

Express to Intra-WCSA. One possible repositioning could involve a vessel leaving the Chennai Express at TPP, and sailing to HKG where it can pick up the Asia-CA3, thereby replacing the on-service vessel. The repositioning vessel would then sail along the Asia-CA3 until it gets to BLB, where it can join the Intra-WCSA. Note that no vessel sails on the backhaul of the Asia-CA3, and this is allowed because very little cargo travels on the Asia-CA3 towards Asia.

2.1 Literature Review

The LSFRP has recently begun to receive attention in the literature, but it was not mentioned in either of the most influential surveys of work in the liner shipping domain [6,7] or container terminals [14]. Although there has been significant work on problems such as the Fleet Deployment Problem (FDP) (e.g., [13]) and the Network Design Problem (NDP) (e.g. [1,11]), these problems deal with strategic decisions related to building the network and assigning vessels to services, rather than the operational problem of finding paths for vessels through the network. Additionally, the vessel schedule recovery problem (VSRP) [3] differs in that it lacks the cost saving activities of the LSFRP due to its short time window. Andersen's PhD thesis [2] discusses a problem similar to the LSFRP, called the Network Transition Problem (NTP), but provides no mathematical model or formal problem description, nor does the problem handle slow steaming, empty equipment flows, or SOS opportunities.

Although tramp shipping problems, such as [5,10], maximize cargo profit while accounting for sailing costs and port fees as in the LSFRP, they lack liner shipping specific constraints, such as sail-on-service opportunities, phase-in requirements and strict visitation times. Airline disruption management (see [8,9]), while also relying on time-based graphs, differs from the LSFRP in that airline disruption management requires an exact cover of all flight legs over a planning horizon. The LSFRP has no such requirement over visitations or sailing legs.

The primary previous work on the LSFRP in the literature is found in [17], [18], [16] and [15], by the authors. The first work on the LSFRP, [17], solved an abstraction of the LSFRP without cargo/equipment flows and SOS parallel sailings using a hybrid of automated planning and linear programming called Linear Temporal Optimization Planning (LTOP). However, LTOP and other automated planning methods are unable to model cargo flows and are thus inapplicable to the version of the LSFRP we solve in this work. A mathematical model of the LSFRP with cargo and equipment flows is introduced in [18], and CPLEX is used to solve the model. In [16], a simulated annealing algorithm is used to solve instances that CPLEX was unable to solve. Finally, in [15] it is shown that the late acceptance hill climbing technique from [4] does not outperform simulated annealing on the full LSFRP. Unlike in [18], [16] and [15], in this work we only investigate a subset of the LSFRP that does not include flexible time windows.

3 Arc Flow Mathematical Model

We describe the mathematical model of the LSFRP with cargo flows from [18] and [16], in which demands are modeled using variables representing the amount of flow of each demand on each arc in the graph.

3.1 Graph Description

We give an overview of the graph used in the model of the IVLSFRP with cargo flows, and refer to [16] for details. The graph is based off of the idea of modeling each visitation as a node in a graph, with arcs representing sailings between visitations. We begin by letting V be the set of visitations for all vessels, and defining S as the set of ships.

The overall structure of the graph involves three sets of visitations: phase-out, phase-in, and SOS visitations. The three types of visitations represent three disjoint sets that make up V. In addition to these visitations, we include a graph sink, τ, which all vessels must reach for a valid repositioning. We let $V' = V \setminus \tau$ be the set of all graph visitations excluding τ. We now describe the arc structure present in each of the three sets of visitations.

Phase-out & Phase-in Each ship is assigned a particular visitation, $\sigma_s \in V'$, at which the ship $s \in S$ begins its repositioning. This visitation represents the earliest allowed phase-out time for that vessel. A visitation is then created for each subsequent port call of the ship on its phase-out slot. Each phase-out visitation is connected to the next one with an arc. Note that phase-out visitations do not connect to the phase-out visitations of other ships.

Vessels may leave phase-out nodes to sail to SOS opportunities or to a phase-in slot. Thus, arcs are created from each phase-out visitation to each phase-in visitation and SOS entry visitation such that sailing between the visitations is temporally feasible (i.e. the starting time of the phase-in/SOS visitation is greater than the end time of the phase-out visitation plus the sailing time). Finally, phase-out visitations have incoming arcs from phase-in visitations in the same trade zone, which we define as a set of ports in the same geographical region. This allows ships to avoid sailing back and forth between ports when transferring directly between the phase-out and phase-in.

We create visitations for each port call along a phase-in slot, and connect subsequent phase-in visitations to each other. The final visitation in a slot, which represents the time at which regular operations must begin on a service, is connected to the graph sink, τ. Due to the node disjointness of the vessel paths, this structure ensures that each slot on the goal service receives a single vessel.

Sail-on-service SOS opportunities are modeled with a special structure that ensures each one is only used by at most a single vessel. The structure partitions the visitations of an SOS into three sets: *entry ports*, where vessels may join the SOS, *through ports*, in which a vessel must already be on the SOS, and *end ports* where a vessel may leave the SOS.

Sailing Cost The fuel consumption of a ship is a cubic function of the speed of the vessel. We precompute the optimal cost for each arc using a linearized bunker consumption function. All arcs in the model are assigned a sailing cost for each ship that is the optimal sailing cost given the total duration of the arc. Since ships have a minimum speed, if the duration of the arc is greater than the time required to sail on the arc at a ship's minimum speed, the cost is calculated using the minimum speed and then the ship simply waits for the remainder of the duration. This is a common practice for shipping lines in order to add buffer to their schedules, thus making the network more robust to disruptions.

3.2 Mixed-Integer Programming Model

We now present the MIP model from [18,16] excluding flexible node/arc components. We use the following parameters and variables for the model.

Parameters

T	Set of equipment types. $T = \{dc, rf\}$.
S	Set of ships.
V'	Set of visitations minus the graph sink, τ.
$\sigma_s \in V'$	Starting visitation of vessel $s \in S$.
V^{t+}	Set of visitations with an equipment surplus of type t.
V^{t-}	Set of visitations with an equipment deficit of type t.
V^{t^*}	Set of visitations with an equipment surplus or deficit of type t ($V^{t^*} = V^{t^+} \cup V^{t^-}$).
$In(i) \subseteq V'$	Set of visitations with an arc connecting to visitation $i \in V$.
$Out(i) \subseteq V'$	Set of visitations receiving an arc from $i \in V$.
$c_i^{Mv} \in \mathbb{R}^+$	Cost of a TEU move at visitation $i \in V'$.
$f_{s,i}^{Port} \in \mathbb{R}$	Port fee associated with vessel s at visitation $i \in V'$.
$r_t^{Eqp} \in \mathbb{R}^+$	Revenue for each TEU of equipment of type $t \in T$ delivered.
$u_s^t \in \mathbb{R}^+$	Capacity of vessel s for cargo type $t \in T$. Note that u_s^{dc} is the capacity of all slots on the vessel, including reefer slots.
A'	Set of arcs $(i, j) \in A$, where $i, j \in V'$.
$c_{i,j}^s$	Fixed cost of vessel s utilizing arc $(i, j) \in A'$.
$(o, d, t) \in \Theta$	A demand triplet, where $o \in V', d \subseteq V'$ and $t \in T$ are the origin visitation, destination visitations and the cargo type, respectively.
$a^{(o,d,t)} \in \mathbb{R}^+$	Amount of demand available for the demand triplet.
$r^{(o,d,t)} \in \mathbb{R}^+$	Amount of revenue gained per TEU for the demand triplet.

Variables

$x^{(o,d,t)}_{i,j} \in [0, a^{(o,d,t)}]$	Amount of flow of demand triplet $(o,d,t) \in \Theta$ on $(i,j) \in A'$.
$x^t_{i,j} \in [0, \max_{s \in S} u^{dc}_s]$	Amount of equipment of type $t \in T$ flowing on $(i,j) \in A'$.
$y^s_{i,j} \in \{0,1\}$	Indicates whether vessel s is sailing on arc $(i,j) \in A$.

Objective and Constraints

$$\max \; -\sum_{s \in S} \sum_{(i,j) \in A'} c^s_{i,j} y^s_{i,j} - \sum_{j \in V'} \sum_{i \in In(j)} \sum_{s \in S} f^{Port}_{s,j} y^s_{i,j} \tag{1}$$

$$+ \sum_{(o,d,t) \in \Theta} \left(\sum_{j \in d} \sum_{i \in In(j)} \left(r^{(o,d,t)} - c^{Mv}_o - c^{Mv}_j \right) x^{(o,d,t)}_{i,j} \right) \tag{2}$$

$$+ \sum_{t \in T} \left(\sum_{i \in V^{t+}} \sum_{j \in Out(i)} \left(r^{Eqp}_t - c^{Mv}_i \right) x^t_{i,j} - \sum_{i \in V^{t-}} \sum_{j \in In(i)} c^{Mv}_i x^t_{j,i} \right) \tag{3}$$

$$\text{s.t.} \; \sum_{s \in S} \sum_{i \in In(j)} y^s_{i,j} \leq 1 \qquad \forall j \in V' \tag{4}$$

$$\sum_{i \in Out(\sigma_s)} y^s_{\sigma_s,i} = 1 \qquad \forall s \in S \tag{5}$$

$$\sum_{i \in In(\tau)} \sum_{s \in S} y^s_{i,\tau} = |S| \tag{6}$$

$$\sum_{i \in In(j)} y^s_{i,j} - \sum_{i \in Out(j)} y^s_{j,i} = 0 \qquad \forall j \in \{V' \setminus \bigcup_{s \in S} \sigma_s\}, s \in S \tag{7}$$

$$\sum_{(o,d,rf) \in \Theta} x^{(o,d,rf)}_{i,j} - \sum_{s \in S} u^{rf}_k y^s_{i,j} \leq 0 \qquad \forall (i,j) \in A' \tag{8}$$

$$\sum_{(o,d,t) \in \Theta} x^{(o,d,t)}_{i,j} + \sum_{t' \in T} x^{t'}_{i,j} - \sum_{s \in S} u^{dc}_s y^s_{i,j} \leq 0 \qquad \forall (i,j) \in A' \tag{9}$$

$$\sum_{i \in Out(o)} x^{(o,d,t)}_{o,i} \leq a^{(o,d,t)} \sum_{i \in Out(o)} \sum_{s \in S} y^s_{o,i} \qquad \forall (o,d,t) \in \Theta \tag{10}$$

$$\sum_{i \in In(j)} x^{(o,d,t)}_{i,j} - \sum_{k \in Out(j)} x^{(o,d,t)}_{j,k} = 0 \qquad \forall (o,d,t) \in \Theta, j \in V' \setminus (o \cup d) \tag{11}$$

$$\sum_{i \in In(j)} x^t_{i,j} - \sum_{k \in Out(j)} x^t_{j,k} = 0 \qquad \forall t \in T, j \in V' \setminus V^{t^*} \tag{12}$$

The domains of the variables are as previously described. The objective consists of four components. First, objective (1) takes into account the precomputed sailing costs on arcs between inflexible visitations and the port fees at each visitation. Note that the fixed sailing cost on an arc does not only include fuel costs, but can also include canal fees or the time-charter bonus for entering an SOS. Second, the profit from delivering cargo (2) is computed based on the revenue from delivering cargo minus the cost to load and unload the cargo from the vessel. Note that the model can choose how much of a demand to deliver, even choosing to deliver a fractional amount. We can allow this since each demand is an aggregation of cargo between two ports, meaning at most one container between two ports will be fractional. Third, equipment profit is taken into account in (3). Equipment is handled similarly to cargo, except that equipment can flow from any port where it is in supply to any port where it is in demand.

Multiple vessels are prevented from visiting the same visitation in constraints (4). The flow of each vessel from its source node to the graph sink is handled by constraints (5), (6) and (7), where (5) starts the vessel flow, (6) ensures that all vessels arrive at the sink, and (7) balances the flow at each node.

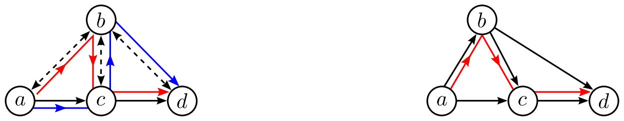
Arcs are assigned a capacity if a vessel utilizes the arc in constraints (8), which assigns the reefer container capacity, and in (9), which assigns the total container capacity, respectively. Note that constraints (8) do not take into account empty reefer equipment, since empty containers do not need to be turned on, and can therefore be placed anywhere on the vessel. Cargo is only allowed to flow on arcs with a vessel in constraints (10). The flow of cargo from its source to its destination, through intermediate nodes, is handled by constraints (11). Constraints (12) balance the flow of equipment in to and out of nodes. Since the amount of equipment carried is limited only by the capacity of the vessel, no flow source/sink constraints are required.

4 Node Flow Model

When the number of demands and arcs grows, the number of variables in the arc flow model can become too large to fit in memory, and even when the model fits in memory, it is still often too large to solve. In both [18] and [16], the authors are unable to solve the LSFRP to optimality on problems with more than 9 vessels. The instances with 9 vessels or more all have a graph of around 10,000 arcs or more, and the number of demands is above 400. Problems of such size require at least four million variables just to model the demand flow, making the problems difficult to solve. In contrast, the number of nodes in the graph is much less than the number of arcs (between 300 and 400 nodes on large instances), meaning a model that can take advantage of flows using knowledge of the nodes used along a path can significantly reduce the number of variables required.

We provide two different node flow based models of the IVLSFRP, each of which uses a different modeling of equipment. In our first model, which we call the *equipment as flows* model, we do not change the way equipment is modeled from the arc flow model. We also provide a model in which we model equipment flows as demands, which we call the *equipment as demands* model.

In order to prevent a vessel from carrying too many containers, the amount of containers loaded on the vessel must be accounted for throughout its repositioning. In the arc flow model, this is done by explicitly keeping track of the amount of demand flowing on each arc. In contrast, the node flow model is able to keep count of the number of containers on the vessel implicitly based on the visitations on a vessel's path, and therefore only needs to determine how many containers from a particular demand are flowing. That is, instead of a variable for each demand on each arc, the node flow model has a variable for each demand on each vessel. In order to ensure a vessel is not overloaded over the course of its repositioning, it suffices to ensure it is not overloaded as it enters each visitation on its path, which corresponds to constraining the incoming arcs of a visitation. Since demands must be loaded and unloaded at visitations, which



(a) A non-fixed time graph.

(b) A fixed time graph.

Fig. 2: Subsets of an LSFRP graph with potential vessel paths (red, blue).

in the IVLSFRP have a fixed begin and end time, the times a demand can be loaded and unloaded are fixed as well. We can therefore determine which demands can be carried on each arc with a reachability analysis.

Since we know in advance which demands can be on the vessel at what times and where, we can post constraints for each visitation to ensure the vessel is not overloaded without explicitly modeling the path of the vessel. These constraints represent the state of a vessel as it enters a visitation, and they neither over or under constrain the problem. They are clearly sufficient to prevent the vessel from being loaded over capacity, since they cover all demands that can possibly be on a vessel as it enters each visitation on its path. They do not over constrain the problem because only those demands which can be loaded on the vessel are constrained. Due to the fixed visitations times, there is never a situation in which two demands are loaded in sequence on one potential vessel path, and are loaded simultaneously on a different path. This means an optimal solution can always be found, if the problem is not infeasible. Consider the following example.

Example 1. Figure 2 shows two graphs. In the first graph (a), node b has no fixed visitation time and must be scheduled, and in the second graph (b), all nodes have a fixed visitation time. A single vessel must sail through the graph, starting at a and ending at d. The demands in the instance are $\Theta = \{(a,c),(b,d)\}$[2]. First, consider the non-fixed time case in Figure 2a. The red path (a,b,c,d) and blue path (a,c,b,d) show two potential voyages of a vessel. On the red path, demand (a,c) is loaded while the vessel is at a, and then the vessel continues to b, where it loads (b,d). Thus, on the red path both demands are loaded on the vessel simultaneously. On the blue path, the vessel first loads the (a,c) demand, sails to c where it is delivered, and continues to b where it loads (b,d). In this case, the demands are loaded sequentially. Now consider the fixed time case in Figure 2b, in which the time when b occurs is known in advance. The red path visits nodes a,b,c,d, and demand (a,c) and (b,d) are on the ship simultaneously at b. In fact, there is no path in the fixed time case where (a,c) and (b,d) can be loaded sequentially; they are either both on the ship or only (a,c) is on the ship.

In the LSFRP, a single demand can be delivered to any visitation in a set of destinations. These destinations all correspond to a single real-world port being visited by different services at different times. The arc flow model takes these multiple destinations into account by simply having each destination visitation of a demand act as a sink for that demand. This is advantageous for the model, since modeling each origin-destination pair individually would require many variables. However, in the node flow

[2] We ignore container types, as they are not relevant.

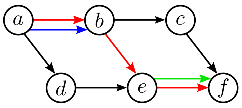


Fig. 3: Sequential demand delivery in a multiple demand destination problem.

model, multiple destinations can cause a situation in which certain incoming arcs of nodes are over-constrained. This could result in the optimal solution not being found.

Example 2. Consider Figure 3, which shows a graph with the vessel path a, b, e, f shown with red, and the demands $\{(a, \{b, f\}), (e, \{f\})\}$. Since it is possible for both demands to be flowing on arc (e, f), a constraint must be posted ensuring that $x^{(a,\{b,f\})} + x^{(e,\{f\})} \leq u_s^{dc}$. When a vessel travels on the path shown, the first demand, rather than being delivered to f, is dropped off at b. The vessel then continues to e where it loads the second demand. Consider the case where the capacity of the vessel is 50 TEU and both demands have 50 TEU available. Due to the constraint we must post on arc (e, f), we can only take a maximum of 50 TEU of both demands, even though is is possible to carry both demands in full.

We remedy this problem by splitting each multiple destination demand into a demand for each origin-destination pair, and add a constraint in the node flow model to ensure that the amount of containers delivered to all of the destinations is not greater than the amount of demand available. In the following models, we use the set Θ' to represent the set of single origin-destination pairs of demands in the following node flow models. Formally, let $\Theta' = \bigcup_{(o,d,t)\in\Theta} \bigcup_{d'\in d}(o, d', t)$.

4.1 Preprocessing

We base our node based model on the same graph and notation as in Section 3. In order to model the flows based on nodes, we must determine which nodes a particular demand can traverse. We do this using a simple reachability analysis based on the transitive closure of the graph. Let $G^T = (V^T, A^T)$ be the transitive closure of the graph $G = (V, A)$, and

$$\Theta_i^{Vis'} = \{(o, d, t) \in \Theta' \mid (o, i) \in A^T \wedge \exists d' \in d \text{ s.t. } \big((i, d') \in A^T \vee i = d'\big)\}.$$

Each node is thereby assigned a set of demands ($\Theta_i^{Vis'}$) based on whether the node is reachable from the demand origin and at least one of the destinations of the demand. We further define $\Theta_{i,rf}^{Vis'} = \{(o, d, t) \in \Theta_i^{Vis'} \mid t = \mathit{rf}\}$ to be the set of reefer demands at node $i \in V'$. Using these sets of demands, we can now model demand flows on the nodes of the IVLSFRP graph.

4.2 Equipment as Flows

We extend the parameters in Section 3.2 with the following three parameters: Θ', the set of single origin single destination demands, $\Theta_i^{Vis'}$, which is the set of demands (dry and reefer) that could traverse node i, and $\Theta_{i,rf}^{Vis'}$, the set of reefer demands that could traverse node i.

Variables

$x_{i,j}^{t} \in [0, \max_{s \in S} u_s^{dc}]$	Amount of equipment of type $t \in T$ flowing on $(i,j) \in A'$.
$y_{i,j}^{s} \in \{0,1\}$	Indicates whether vessel s is sailing on arc $(i,j) \in A$.
$z_s^{(o,d,t)} \in [0, a^{(o,d,t)}]$	Amount of demand triplet $(o,d,t) \in \Theta'$ carried on ship $s \in S$.

Objective and Constraints

$$\max \; -(1) + (3) + \sum_{s \in S} \sum_{(o,d,t) \in \Theta'} \left(r^{(o,d,t)} - c_o^{Mv} - c_d^{Mv} \right) z_s^{(o,d,t)} \tag{13}$$

s. t. (4); (5); (6); (7); (12);

$$z_s^{(o,d,t)} \leq a^{(o,d,t)} \sum_{i \in Out(o)} y_{o,i}^{s} \qquad \forall (o,d,t) \in \Theta', s \in S \tag{14}$$

$$z_s^{(o,d,t)} \leq a^{(o,d,t)} \sum_{d' \in d} \sum_{i \in In(d')} y_{id'}^{s} \qquad \forall (o,d,t) \in \Theta', s \in S \tag{15}$$

$$\sum_{(o,d,t) \in \Theta_i^{Vis'}} z_s^{(o,d,t)} + \sum_{t \in T} \sum_{j \in In(i)} x_{i,j}^{t} \leq u_s^{dc} \qquad \forall s \in S, i \in V' \tag{16}$$

$$\sum_{(o,d,t) \in \Theta_{i,rf}^{Vis'}} z_s^{(o,d,t)} \leq u_s^{rf} \qquad \forall s \in S, i \in V' \tag{17}$$

$$\sum_{t \in T} x_{i,j}^{t} \leq \sum_{s \in S} u_s^{dc} y_{i,j}^{s} \qquad \forall (i,j) \in A' \tag{18}$$

$$\sum_{d \in d'} z_s^{(o,d',t)} \leq a^{(o,d,t)} \qquad \forall s \in S, (o,d,t) \in \Theta'. \tag{19}$$

The objective (13) contains the same calculation of sailing costs, equipment profits, and port fees as in the arc flow model. However, the demand profit is now computed using the demand flow variables. Note that unlike in Θ, all $(o,d,t) \in \Theta'$ have $|d| = 1$. Thus, the constant c_d^{Mv} refers to the cost at a particular visitation.

We copy constraints (4) through (7) and (12) directly from the arc flow model in order to enforce node disjointness along vessel paths and create the vessel and equipment flows. We refer readers to Section 3.2 for a full description of these constraints.

Constraints (14) and (15) allow a demand to be carried only if a particular vessel visits both the origin and a destination of the demand. Note that we do not need to limit the demands to be taken only by a single vessel because of the node disjointness enforced by constraints (4). Only a single vessel can enter the origin node of a demand, ensuring that only one vessel can carry a demand.

In constraints (16) and (17) we ensure that the capacity of the vessel is not exceeded at any node in the graph in terms of all containers and reefer containers, respectively. Equipment flows are handled in the dry capacity constraints (16). Due to the equipment balance constraints, ensuring that the equipment capacity is not exceeded at a node is sufficient for ensuring that the vessel is not overloaded.

Constraints (18) prevent equipment from flowing on any arc that does not have a ship sailing on it. When a vessel utilizes an arc, the constraint allows as much equipment to flow as the capacity of the vessel. When an arc has no vessel, the corresponding equipment flow variables have an upper bound of 0. And, finally, constraints (19) ensure that the amount of demand carried for each single origin-destination demand does not exceed the amount of containers that are actually available.

4.3 Equipment as Demands

As an alternative to modeling equipment as flows, we present a model that creates demands for each equipment pair and adds them to Θ'. We let $\Theta_t^{E'} = \{(o, \{d\}, dc) \mid o \in V^{t+} \wedge d \in V^{t-}\}$ be the set of demands corresponding to every pair of visitations with an equipment surplus/deficit for type $t \in T$. Note that we set the demand type of all of the equipment demands to be dry (dc). We then append the equipment demands to Θ' as follows: $\Theta' \leftarrow \Theta' \cup \bigcup_{t \in T} \Theta_t^{E'}$. In addition, let $a^{(o,d,dc)} = \sum_{s \in S} u_s^{dc}$ and $r^{(o,d,dc)} = r_{dc}^{Eqp}$ for all $t \in T, (o, d, dc) \in \Theta_t^{E'}$. Thus, the maximum amount of equipment available for each equipment demand is equal to the sum of the capacities of all ships, and the revenue per TEU delivered is equal to the equipment revenue for each equipment type. Our model uses the same parameters as the arc flow model in Section 3 and the equipment as flow model in Section 4.2. The majority of the model is the same as the previous two models, however we include all of the objectives and constraints for completeness.

Objective and Constraints We using the variables $y_{i,j}^s$ and $z_s^{(o,d,t)}$ from the equipment as flows model and require no additional variables. The model is as follows:

$$\max -(1) + (13) \tag{20}$$

subject to constraints (4), (5), (6), (7), from the arc flow model in Section 3.2, and (14), (15), (17), and (19) from the equipment as flows model in Section 4.2. In place of the dry capacity constraint in the equipment as flows model, we provide the following constraint:

$$\sum_{(o,d,t) \in \Theta_i^{Vis'}} z_s^{(o,d,t)} \leq u_s^{dc} \qquad \forall s \in S, i \in V'. \tag{21}$$

The objective, (20), combines the sailing costs and port fees from the arc flow model with the cargo demand objective from the equipment as demands model. Note the lack of any objective relating to equipment, as it is now a part of the demand structure.

As in the equipment as flows model, we include several constraints from the arc flow model to enforce node disjointness along vessel paths and control the vessel flows. However, we omit the equipment flow balance constraints (12). We also include the node demand constraints from the equipment as flows model, along with the reefer capacity constraint, as they are unaffected by modeling equipment as demands. We modify the dry capacity constraints (16) to produce constraints (21), in which the sum of the demands carried at a particular node must respect the vessel capacity.

5 Computational Evaluation

We evaluated our node flow model, with both versions of equipment handling, on the 37 confidential and 37 public LSFRP instances where only inflexible visitations are present [16]. The instances include two real-world repositioning scenarios as well as a number of crafted scenarios based on data from Maersk Line. Table 1 shows the results of running the arc flow model (AF), equipment as flows (EAF), and equipment as demands (EAD) models on the confidential instance set[3] with a timeout of 5 hours of CPU

[3] The missing instance IDs correspond to instances with flexible arcs, which the EAD and EAF models do not solve.

time and a maximum of 10 GB of memory. We used CPLEX 12.4 on AMD Opteron 2425 HE processors. The table's columns describe the following instance information. Each instance has $|S|$ ships, $|V|$ is the number of visitations, $|A|$ is the number of arcs, $|\Theta|$ is the number of demands, $|E| = |\bigcup_{t \in T} V^{t^*}|$ is the number of visitations with either an equipment surplus or deficit, and $|SOS|$ is the number of SOS structures in the graph. The EAD model is able to solve all but three of the confidential IVLSFRP instances within 5 hours of CPU time, while the EAF model solves 3 previously unsolved instances. Additionally, for instances with between one and eight ships, the CPU time is significantly improved, with an average decrease in CPU time of 138 seconds for both EAF and EAD over AF. The largest speed improvement is on instance repo30c, where EAF and EAD are roughly 500 times faster than AF.

Even on the instances where EAD or EAF timeout, CPLEX is able to provide feasible solutions, unlike in the case of the AF model, where no feasible solution is found for any instance that exceeds the memory allotment or CPU timeout. The EAD model is able to find a solution with a 10.16% gap (to the LP relaxation) for repo42c, a 96.41% gap on repo43c, and an 88.96% gap on repo44c. Although EAD requires over an hour to solve several instances, it finds the optimal solution within an hour on both repo37c and repo40c, but requires extra time to prove optimality. On repo39c and repo40c, EAD is able to find an optimality gap of 8.31% and 10.07% within an hour, respectively.

Table 2 gives our results for the public instances in the IVLSFRP dataset. The columns of the table are identical to Table 1. As in the case of the confidential instances, the node flow model achieves significant performance gains, both in the EAF and EAD cases. With the EAD model, the node flow model is able to provide an optimal answer to every instance in the datset. The largest time improvements are on instances repo27p, repo28p and repo29p, where both the EAD and EAF models allow CPLEX to find the optimal solution in under a second, but the AF model requires over 15 minutes.

6 Conclusion

We introduced a novel, node flow based mathematical model for the IVLSFRP, a fundamental problem to the operations of the world's shipping lines. We provided two versions of the node flow model, in which we modeled equipment flows using an arc flow approach, as well as by converting the equipment flows into demands. Both versions of the model showed significant speedups over the arc flow model, and were able to solve IVLSFRP instances to optimality in CPLEX that the arc flow model cannot even load into memory. In addition, our node flow model offers multiple orders of magnitude improvements on a number of instances in the LSFRP dataset. For future work, we will try to apply the ideas from the node flow model to the full LSFRP.

7 Acknowledgements

We would like to thank our industrial collaborators Mikkel Muhldorff Sigurd and Shaun Long at Maersk Line for their support and detailed description of the fleet repositioning problem. This research is sponsored in part by the Danish Council for Strategic Research as part of the ENERPLAN research project.

ID	$\|S\|$	$\|V\|$	$\|A\|$	$\|\Theta\|$	$\|E\|$	$\|SOS\|$	AF	EAF	EAD
repo1c	3	30	94	27	0	1	0.05	0.48	0.04
repo2c	3	30	94	27	0	2	0.04	0.04	0.04
repo3c	3	37	113	25	0	2	0.03	0.03	0.03
repo4c	3	40	143	21	0	3	0.03	0.37	0.03
repo5c	3	47	208	24	0	3	0.05	0.02	0.03
repo6c	3	47	208	24	0	3	0.05	0.04	0.03
repo7c	3	53	146	51	0	4	0.07	0.04	0.04
repo10c	4	58	389	150	0	0	15.98	0.34	0.33
repo12c	4	74	469	174	0	2	93.65	0.88	0.86
repo13c	4	80	492	186	0	4	175.32	0.87	0.82
repo14c	4	80	492	186	24	4	127.48	0.93	1.05
repo15c	5	68	237	214	0	0	0.37	0.23	0.23
repo16c	5	103	296	396	0	5	0.9	0.36	0.35
repo17c	6	100	955	85	0	0	5.09	0.73	0.71
repo18c	6	133	1138	101	0	9	6.77	0.79	0.75
repo19c	6	133	1138	101	33	9	7.25	0.87	0.98
repo20c	6	140	1558	97	0	4	262.21	1.45	1.41
repo21c	6	140	1558	97	13	4	53.95	1.62	1.55
repo22c	6	140	1558	97	37	4	94.46	1.19	1.65
repo24c	7	75	395	196	0	3	2.44	0.35	0.32
repo25c	7	77	406	195	0	0	2.64	0.32	0.30
repo26c	7	77	406	195	16	0	1.95	0.41	0.34
repo27c	7	78	502	237	0	0	97.12	0.53	0.51
repo28c	7	89	537	241	0	4	174.44	0.66	0.55
repo29c	7	89	537	241	19	4	126.97	0.62	0.59
repo30c	8	126	1154	165	0	0	2058.45	3.76	4.15
repo31c	8	126	1300	312	0	0	105.49	4.93	4.76
repo32c	8	140	1262	188	0	3	494.39	6.97	7.12
repo34c	9	304	9863	435	0	0	CPU	2256.99	2532.11
repo36c	9	364	11078	1280	0	4	Mem	CPU	16203.26
repo37c	9	371	10416	1270	114	7	Mem	7033.32	6330.48
repo39c	9	379	10660	1371	0	7	Mem	7125.89	7142.55
repo40c	9	379	10660	1371	118	7	Mem	15857.61	10049.79
repo41c	10	249	7654	473	0	0	CPU	2543.09	3954.17
repo42c	11	275	5562	1748	0	5	CPU	CPU	CPU
repo43c	11	320	11391	1285	0	0	Mem	CPU	CPU
repo44c	11	328	11853	1403	0	4	Mem	CPU	CPU

Table 1: Time required to solve the arc flow (AF), equipment as flows (EAF), and equipment as demands (EAD) models to optimality in CPLEX 12.4, along with instance statistics for all of the confidential instances with no flexible arcs from [16].

ID	$\|S\|$	$\|V\|$	$\|A\|$	$\|\Theta\|$	$\|E\|$	$\|SOS\|$	AF	EAF	EAD
repo1p	3	36	150	28	0	1	0.06	0.08	0.06
repo2p	3	36	150	28	0	2	0.06	0.05	0.05
repo3p	3	38	151	24	0	2	0.04	0.04	0.04
repo4p	3	42	185	20	0	3	0.05	0.23	0.03
repo5p	3	51	270	22	0	3	0.07	0.05	0.05
repo6p	3	51	270	22	0	3	0.07	0.06	0.05
repo7p	3	54	196	46	0	4	0.08	0.05	0.05
repo10p	4	58	499	125	0	0	74.87	0.27	0.26
repo12p	4	74	603	145	0	2	132.8	0.31	0.28
repo13p	4	80	632	155	0	4	101	0.33	0.31
repo14p	4	80	632	155	24	4	171.79	0.35	0.33
repo15p	5	71	355	173	0	0	0.47	0.31	0.29
repo16p	5	106	420	320	0	5	1.09	0.39	0.38
repo17p	6	102	1198	75	0	0	4.59	1.01	0.94
repo18p	6	135	1439	87	0	9	6.52	0.95	0.82
repo19p	6	135	1439	87	33	9	5.67	1.03	1.08
repo20p	6	142	1865	80	0	4	13.68	1.44	1.15
repo21p	6	142	1865	80	13	4	16.24	2.06	1.36
repo22p	6	142	1865	80	37	4	19.13	1.69	1.44
repo24p	7	75	482	154	0	3	1.65	0.37	0.34
repo25p	7	77	496	156	0	0	3.95	0.41	0.38
repo26p	7	77	496	156	16	0	3.74	0.47	0.40
repo27p	7	79	571	188	0	0	1265.98	0.57	0.50
repo28p	7	90	618	189	0	4	1319.39	0.55	0.46
repo29p	7	90	618	189	19	4	1039.57	0.57	0.54
repo30p	8	126	1450	265	0	0	286.7	13.10	12.60
repo31p	8	130	1362	152	0	0	23.38	30.88	28.89
repo32p	8	144	1501	170	0	3	50.95	45.99	41.46
repo34p	9	304	10577	344	0	0	CPU	7652.67	7388.75
repo36p	9	364	11972	1048	0	4	Mem	CPU	CPU
repo37p	9	371	11371	1023	114	7	Mem	1408.75	790.74
repo39p	9	379	11666	1109	0	7	Mem	1701.53	1911.40
repo40p	9	379	11666	1109	118	7	Mem	3178.03	1859.09
repo41p	10	249	8051	375	0	0	CPU	659.75	727.78
repo42p	11	279	6596	1423	0	5	CPU	4930.85	4006.27
repo43p	11	320	13058	1013	0	0	Mem	CPU	CPU
repo44p	11	328	13705	1108	0	4	Mem	CPU	CPU

Table 2: Time required to solve the arc flow (AF), equipment as flows (EAF), and equipment as demands (EAD) models to optimality in CPLEX 12.4, along with instance statistics for all of the public instances with no flexible arcs from [16].

References

1. J.F. Álvarez. Joint routing and deployment of a fleet of container vessels. *Maritime Economics and Logistics*, 11(2):186–208, June 2009.
2. M.W. Andersen. *Service Network Design and Management in Liner Container Shipping Applications*. PhD thesis, Technical University of Denmark, Department of Transport, 2010.
3. B.D. Brouer, J. Dirksen, D. Pisinger, C.E.M Plum, and B. Vaaben. The Vessel Schedule Recovery Problem (VSRP) – A MIP model for handling disruptions in liner shipping. *European Journal of Operational Research*, 224(2):362–374, 2013.
4. E.K. Burke and Y. Bykov. A late acceptance strategy in hill-climbing for exam timetabling problems. In *PATAT 2008 Conference, Montreal, Canada*, 2008.
5. M. Christiansen. Decomposition of a combined inventory and time constrained ship routing problem. *Transportation Science*, 33(1):3–16, 1999.
6. M. Christiansen, K. Fagerholt, B. Nygreen, and D. Ronen. Maritime transportation. *Handbooks in operations research and management science*, 14:189–284, 2007.
7. M. Christiansen, K. Fagerholt, and D. Ronen. Ship routing and scheduling: Status and perspectives. *Transportation Science*, 38(1):1–18, 2004.
8. J. Clausen, A. Larsen, J. Larsen, and N.J. Rezanova. Disruption management in the airline industry–concepts, models and methods. *Computers & Operations Research*, 37(5):809–821, 2010.
9. N. Kohl, A. Larsen, J. Larsen, A. Ross, and S. Tiourine. Airline disruption management–perspectives, experiences and outlook. *Journal of Air Transport Management*, 13(3):149–162, 2007.
10. J.E. Korsvik, K. Fagerholt, and G. Laporte. A large neighbourhood search heuristic for ship routing and scheduling with split loads. *Computers & Operations Research*, 38(2):474 – 483, 2011.
11. B. Løfstedt, J.F. Alvarez, C.E.M. Plum, D. Pisinger, and M.M. Sigurd. An integer programming model and benchmark suite for liner shipping network design. Technical Report 19, DTU Management, 2010.
12. J. Meyer, R. Stahlbock, and S. Voß. Slow steaming in container shipping. In *Proceedings of the 45th Hawaii International Conference on System Science (HICSS)*, pages 1306–1314. IEEE, 2012.
13. B.J. Powell and A.N. Perakis. Fleet deployment optimization for liner shipping: An integer programming model. *Maritime Policy and Management*, 24(2):183–192, Spring 1997.
14. R. Stahlbock and S. Voß. Operations research at container terminals: a literature update. *OR Spectrum*, 30(1):1–52, 2008.
15. K. Tierney. Late Acceptance Hill Climbing for the Liner Shipping Fleet Repositioning Problem. In *Proceedings of the 14th EU/ME Workshop*, pages 21–27, 2013.
16. K. Tierney, B. Áskelsdóttir, R.M. Jensen, and D. Pisinger. Solving the liner shipping fleet repositioning problem with cargo flows. Technical Report TR-2013-165, IT University of Copenhagen, January 2013.
17. K. Tierney, A.J. Coles, A.I. Coles, C. Kroer, A.M Britt, and R.M. Jensen. Automated planning for liner shipping fleet repositioning. In L. McCluskey, B. Williams, J.R. Silva, and B. Bonet, editors, *Proceedings of the 22nd International Conference on Automated Planning and Scheduling*, pages 279–287, 2012.
18. K. Tierney and R. M. Jensen. The Liner Shipping Fleet Repositioning Problem with Cargo Flows. In Hao Hu, Xiaoning Shi, Robert Stahlbock, and Stefan Voß, editors, *Computational Logistics*, volume 7555 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2012.