Multi-objective Optimization for Liner Shipping Fleet Repositioning

Kevin Tierney[1(✉)], Joshua Handali[2], Christian Grimme[2], and Heike Trautmann[2]

[1] Decision Support and Operations Research Lab, University of Paderborn, Paderborn, Germany
tierney@dsor.de
[2] Information Systems and Statistics Group, University of Münster, Münster, Germany
j_hand02@uni-muenster.de,
{christian.grimme,heike.trautmann}@wi.uni-muenster.de

Abstract. The liner shipping fleet repositioning problem (LSFRP) is a central optimization problem within the container shipping industry. Several approaches exist for solving this problem using exact and heuristic techniques, however all of them use a single objective function for determining an optimal solution. We propose a multi-objective approach based on a simulated annealing heuristic so that repositioning coordinators can better balance profit making with cost-savings and environmental sustainability. As the first multi-objective approach in the area of liner shipping routing, we show that giving more options to decision makers need not be costly. Indeed, our approach requires no extra runtime than a weighted objective heuristic and provides a rich set of solutions along the Pareto front.

1 Introduction

Liner shipping networks form the backbone of world trade, providing efficient, cheap and secure transportation of containerized goods. The container shipping industry carried over 171 million twenty foot equivalent units (TEU)[1] in 2015 on over 5,000 ships [18]. As container volumes steadily increase each year, the challenge of maintaining liner shipping networks becomes ever more difficult for liner carriers to handle.

Liner shipping networks consist of regularly scheduled, cyclical routes between ports, called *services*. Services are regularly added, removed and modified in a network to adapt to seasonal cargo demands and macro economic changes. To carry out such changes, ships must be moved, or *repositioned*, between services. Liner carriers must balance the conflicting objectives of minimizing their sailing costs and maximizing their cargo intake (profit). Carriers

[1] A single TEU represents one twenty foot container, with two TEU representing the commonly found forty foot container.

H. Trautmann et al. (Eds.): EMO 2017, LNCS 10173, pp. 622–638, 2017.
DOI: 10.1007/978-3-319-54157-0_42

are particularly interested in reducing their carbon footprint through improved routing and slow-steaming (see, e.g., [9]), which also results in reduced costs due to lower fuel consumption.

Experienced planners create repositioning plans often with little to no decision support [17]. Recently, however, several algorithms have been developed for solving the liner shipping fleet repositioning problem (LSFRP), both to optimality [12,13,15] and heuristically [12,14]. In addition, a decision support system (DSS) with an integrated repositioning visualization has been developed [10] to assist planners making repositioning decisions. However, all of the research in this area is based on a single objective function consisting of weighted costs and revenues. The solutions found with this function may not be desirable for liner carriers. Moreover, trade-offs between objectives are ignored and cannot be adequately assessed by the decision maker.

We propose a multi-objective simulated annealing (MOSA) algorithm for the LSFRP that allows repositioning coordinators to find LSFRP solutions under varying sailing costs, cargo profit and empty container transports. Our algorithm provides solutions across a wide expanse of the Pareto front, ensuring that coordinators can easily choose the solution that best fits the current business environment. We provide experimental results and a brief case study on LSFRP instances based on data from Maersk Line, the world's largest liner carrier.

This paper is organized as follows. We first briefly describe fleet repositioning in Sect. 2, followed by an overview of related work in Sect. 3. Next, we describe MOSA for the LSFRP in Sect. 4 along with detailed experimental results in Sect. 5. Finally, we conclude and discuss future work in Sect. 6.

2 Liner Shipping Fleet Repositioning

We briefly describe the LSFRP and provide a mathematical model, referring readers to [17] for a more detailed description of the problem and the liner shipping domain.

The process of repositioning a set of container ships has several phases. Before a repositioning begins, ships sail along their regularly scheduled route(s). In contrast to other forms of shipping, such as tramp or industrial, liner shipping is completely schedule based, like a bus or train, with ships arriving at ports on a periodic (usually weekly or bi-weekly) basis. At a time point determined by the repositioning coordinator the vessel *phases out* from its service and begins repositioning towards its new service, called the *goal service*. During this time, the vessel may undertake any number of activities to complete its repositioning as described below. Once the vessel reaches its goal service, it *phases in*, which indicates the start of regular service.

Vessel repositioning is expensive due to the cost of fuel (on the order of hundreds of thousands of dollars) and the revenue lost due to cargo flow disruptions. Liner carriers reposition hundreds of vessels each year, meaning that optimizing vessel movements can significantly reduce the economic and environmental burdens of containerized shipping, and furthermore allow shippers to better utilize repositioning vessels to transport cargo. The aim of the LSFRP is to maximize the profit earned when repositioning a number of vessels from their initial services to a goal service being added or expanded.

Liner Shipping Services. Services are composed of multiple *slots*, each of which represents a cycle assigned to a particular vessel. A slot consists of a number of *visits*, or 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 visit in the slot.

Vessel sailing speeds can be adjusted throughout repositioning to balance cost savings with punctuality, as vessel fuel consumption increases approximately cubically with the speed of the vessel. *Slow steaming*, in which vessels sail near or at their minimum speed, allows vessels to sail more cheaply between two ports than at higher speeds, albeit with a longer duration. We linearize the bunker consumption of each repositioning vessel in order to more easily model the LSFRP.

Phase-Out. The repositioning period for each vessel starts at a specific time when the vessel may cease normal operations. Each vessel is assigned a different starting time, or phase-out time, for its repositioning. The repositioning coordinator specifies these times based on customer requirements, leasing agreements, etc. However, the phase out time does not require a ship to leave its service immediately. Ships may continue sailing on their initial service if it is profitable to do so in the context of the required repositioning. Note that the ships involved in a repositioning often do not all share the same initial service. After ships leave their initial service, they may undertake a number of repositioning activities.

Phase-In. Each vessel must phase in to a slot on the goal service before a time set by the repositioning coordinator. After this time, normal operations on the goal service must begin. Some services start with all ships arriving one week after the other at the same port, while large services may begin simultaneously in multiple ports. The repositioning of each vessel and optimization of its activities thus 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 take a detailed view of cargo flows in which a demand is represented with a source port, destination port, container type, latest delivery time, amount of TEU available and revenue per TEU delivered. There are typically multiple customers shipping containers between a given pair of two ports, all at different prices due to differing agreements between the carrier and the shipper. We aggregate all of these demands into a single demand and compute an average revenue per container for the port to port pair. Furthermore, we do not require that all demands are delivered. Demands can be ignored or partially delivered with no penalty.

For each TEU of cargo transported, the liner carrier must pay for the cost of loading and unloading the containers from the ship from the revenue to determine the profit per TEU of each demand. Vessels have a maximum capacity in terms of both *dry* and *reefer* (refrigerated) cargo. Reefer containers must be placed in positions with plugs or refrigeration hookups, which are limited on each vessel. Dry containers can be placed anywhere, however if one is placed in a reefer position it prevents a reefer container from being loaded there. We note that container stowage is in its own right a difficult optimization problem [16], which is why we only constrain the overall capacity of the vessel. The cargo flows can be seen as a multi-commodity flow where each demand is a commodity with a start and end port. Equipment, in contrast to cargo, can be sent from any port where it is in surplus to any port where it is in demand and the revenue represents the cost saved by the carrier by moving the equipment through a repositioning rather than a more expensive method.

Some ports have equipment, but are not on any service visited by repositioning vessels. These ports are called *flexible* ports, and are associated with flexible visits. The repositioning coordinator may choose the time a vessel arrives at such visits, if at all. All other visits are called *inflexible*, because the time a vessel arrives is fixed.

Sail-on-Service (SoS) Opportunities. While repositioning, vessels may use certain services to cheaply sail between different parts of the network called *SoS opportunities*. There are two vessels involved in SoS opportunities, 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 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 fuel, since the on-service vessel no longer has to sail anywhere and can be laid up or leased out. Utilizing an SoS is subject to a number of constraints, which are described in [17].

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 Express to Intra-WCSA. One possible repositioning plan (marked in red) could involve a vessel leaving the Chennai Express at TPP, and sailing to HKG, where it can pick up the Asia-CA3, thus 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.

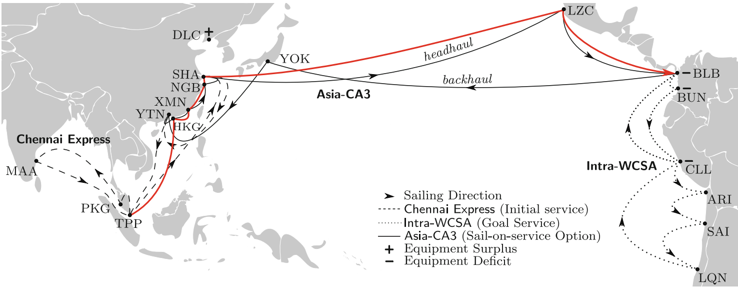


Fig. 1. A subset of a case study of the LSFRP, from [13].

2.1 Mathematical Model

We present the mixed-integer linear arc-flow formulation of the LSFRP from [12]. Several other models exist for special cases of the LSFRP that are detailed in [17].

Graph. The LSFRP can be modeled using a graph in which each visit is represented by a node, and valid sailings between visits are arcs between nodes in the graph. This means that graph nodes represent both a place and a time, except in the case of flexible nodes, where the time a ship arrives must be determined. We note that the LSFRP graph is rather detailed and embeds several LSFRP constraints (see [17]).

Parameters. The parameters of the model are as follows.

S	Set of ships.
V'	Set of visits minus the graph sink.
V^i, V^f	Set of inflexible and flexible visits, respectively.
A^i, A^f	Set of inflexible and flexible arcs, respectively.
A'	Set of arcs minus those arcs connecting to the graph sink, i.e., $(i, j) \in A$, $i, j \in V'$.
Q	Set of equipment types. $Q = \{dc, rf\}$.
M	Set of demand triplets of the form (o, d, q), where $o \in V', d \subseteq V'$ and $q \in Q$ are the origin visit, destination visits and the cargo type, respectively.
$V^{q+} \subseteq V'$	Set of visits with an equipment surplus of type q.
$V^{q-} \subseteq V'$	Set of visits with an equipment deficit of type q.
$V^{q^*} \subseteq V'$	Set of visits with an equipment surplus or deficit of type q ($V^{q^*} = V^{q^+} \cup V^{q^-}$).
$u_s^q \in \mathbb{R}^+$	Capacity of vessel s for cargo type $q \in Q$.
$M_i^{Orig}, (M_i^{Dest}) \subseteq M$	Set of demands with an origin (destination) visit $i \in V$.
$v_s \in V'$	Starting visit of ship $s \in S$.
$t_{si}^{Mv} \in \mathbb{R}$	Move time per TEU for vessel s at visit $i \in V'$.

$t_i^E \in \mathbb{R}$	Enter time at inflexible visit $i \in V'$.
$t_i^X \in \mathbb{R}$	Exit time at inflexible visit $i \in V'$.
$t_i^P \in \mathbb{R}$	Pilot time at visit $i \in V'$.
$r^{(o,d,q)} \in \mathbb{R}^+$	Amount of revenue gained per TEU for the demand triplet.
$c_{sij}^{Sail} \in \mathbb{R}^+$	Fixed cost of vessel s utilizing arc $(i,j) \in A'$.
$c_{sij}^{VarSail} \in \mathbb{R}^+$	Variable hourly cost of vessel $s \in S$ utilizing arc $(i,j) \in A'$.
$c_i^{Mv} \in \mathbb{R}^+$	Cost of a TEU move at visit $i \in V'$.
$c_{si}^{Port} \in \mathbb{R}$	Port fee associated with vessel s at visit $i \in V'$.
$d_{ijs}^{Min} \in \mathbb{R}^+$	Minimum duration for vessel s to sail on flexible arc (i,j).
$d_{ijs}^{Max} \in \mathbb{R}^+$	Maximum duration for vessel s to sail on flexible arc (i,j).
$a^{(o,d,q)} \in \mathbb{R}^+$	Amount of demand available for the demand triplet.
$In(i) \subseteq V'$	Set of visits with an arc connecting to visit $i \in V$.
$Out(i) \subseteq V'$	Set of visits receiving an arc from $i \in V$.
$\tau \in V$	Graph sink, which is not an actual visit.
ϕ^{Sail}, ϕ^{Cargo}, ϕ^{Eqp}	Weights for the objective functions.

Variables. We now list the decision variables of the model.

$w_{ij}^s \in \mathbb{R}_0^+$	The duration that vessel $s \in S$ sails on flexible arc $(i,j) \in A^f$
$x_{ij}^{(o,d,q)} \in \mathbb{R}_0^+$	Amount of flow of demand triplet $(o,d,q) \in M$ on $(i,j) \in A'$
$x_{ij}^q \in \mathbb{R}_0^+$	Amount of equipment of type $q \in Q$ flowing on $(i,j) \in A'$
$y_{ij}^s \in \{0,1\}$	Indicates whether vessel s is sailing on arc $(i,j) \in A$
$z_i^E \in \mathbb{R}_0^+$	Defines the enter time of a vessel at visit i
$z_i^X \in \mathbb{R}_0^+$	Defines the exit time of a vessel at visit i.

Objective and Constraints

$$\max \; -\phi^{Sail}\Bigg[\sum_{s\in S}\left(\sum_{(i,j)\in A'} c_{sij}^{Sail} y_{ij}^s + \sum_{(i,j)\in A^f} c_{sij}^{VarSail} w_{ij}^s\right)$$

$$+ \sum_{j\in V'}\sum_{i\in In(j)}\sum_{s\in S} c_{sj}^{Port} y_{ij}^s\Bigg] \tag{1}$$

$$+ \phi^{Cargo}\sum_{(o,d,q)\in M}\left(\sum_{j\in d}\sum_{i\in In(j)}\left(r^{(o,d,q)} - c_o^{Mv} - c_j^{Mv}\right)x_{ij}^{(o,d,q)}\right) \tag{2}$$

$$+ \phi^{Eqp}\sum_{q\in Q}\sum_{i\in V^{q+}}\sum_{j\in Out(i)} x_{ij}^q \tag{3}$$

$$\text{subject to} \sum_{s \in S} \sum_{i \in In(j)} y_{ij}^s \leq 1 \qquad \forall j \in V' \tag{4}$$

$$\sum_{j \in Out(i)} y_{ij}^s = 1 \qquad \forall s \in S, i = v_s \tag{5}$$

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

$$\sum_{i \in In(j)} y_{ij}^s - \sum_{i \in Out(j)} y_{ji}^s = 0 \qquad \forall j \in \{V' \setminus \bigcup_{s \in S} v_s\}, s \in S \tag{7}$$

$$\sum_{(o,d,rf) \in M} x_{ij}^{(o,d,rf)} \leq \sum_{s \in S} u_s^{rf} y_{ij}^s \qquad \forall (i,j) \in A' \tag{8}$$

$$\sum_{(o,d,q) \in M} x_{ij}^{(o,d,q)} + \sum_{q' \in Q} x_{ij}^{q'} \leq \sum_{s \in S} u_s^{dc} y_{ij}^s \quad \forall (i,j) \in A' \tag{9}$$

$$\sum_{i \in Out(o)} x_{oi}^{(o,d,q)} \leq a^{(o,d,q)} \sum_{i \in Out(o)} \sum_{s \in S} y_{oi}^s \quad \forall (o,d,q) \in M \tag{10}$$

$$\sum_{i \in In(j)} x_{ij}^{(o,d,q)} - \sum_{k \in Out(j)} x_{jk}^{(o,d,q)} = 0 \ \forall (o,d,q) \in M, j \in V' \setminus (o \cup d) \tag{11}$$

$$\sum_{i \in In(j)} x_{ij}^q - \sum_{k \in Out(j)} x_{jk}^q = 0 \qquad \forall q \in Q, j \in V' \setminus V^{q^*} \tag{12}$$

$$d_{ijs}^{Min} y_{ij}^s \leq w_{ij}^s \leq d_{ijs}^{Max} y_{ij}^s \qquad \forall (i,j) \in A^f, s \in S \tag{13}$$

$$z_i^E = t_i^E \sum_{s \in S} \sum_{j \in In(i)} y_{ij}^s \qquad \forall i \in V^i \tag{14}$$

$$z_i^X = t_i^X \sum_{s \in S} \sum_{j \in Out(i)} y_{ij}^s \qquad \forall i \in V^i \tag{15}$$

$$z_i^X + \sum_{s \in S} w_{ij}^s \leq z_j^E \qquad \forall (i,j) \in A^f \tag{16}$$

$$\begin{aligned} &\sum_{(o,d,q) \in M_i^{Orig}} \sum_{j \in Out(o)} t_{so}^{Mv} x_{oj}^{(o,d,q)} + \sum_{(o,d,q) \in M_i^{Dest}} \sum_{d' \in d} \sum_{j \in In(d')} t_{sd}^{Mv} x_{jd'}^{(o,d,q)} \\ &+ \sum_{q \in Q} \left(\sum_{i' \in \{V^{q+} \cap \{i\}\}} \sum_{j \in Out(i')} t_{si'}^{Mv} x_{i'j}^q + \sum_{i' \in \{V^{q-} \cap \{i\}\}} \sum_{j \in In(i')} t_{sj}^{Mv} x_{ji'}^q \right) \\ &- z_i^X + z_i^E + t_i^P \sum_{j \in In(i)} y_{ij}^s \leq 0 \qquad \forall i \in V^f, s \in S \end{aligned} \tag{17}$$

The domains of the variables are described above. The objective function consists of three components. The sailing cost objective (1) takes into account the precomputed sailing costs on arcs between inflexible visits, the variable cost for sailings to and from flexible visits, and port fees. Note that the fixed sailing cost on an arc may also include canal fees or other revenue related to SoS opportunities. The profit from delivering cargo (2) earns revenue from delivering cargo subtracted by the cost to load and unload the cargo from the vessel. The equipment profit is taken into account in (3). Equipment is handled similar to cargo, except that equipment can flow from any port where it is in supply to any port where it is in demand. When $\phi^{Cargo} = \phi^{Sail} = 1$ and $\phi^{Equip} = 150$ the objective function is nearly equivalent to the one used in previous work on the LSFRP.

The node-disjointness of the paths is enforced by Constraints (4), and the flow of each vessel from its source node to the graph sink is handled by (5), (6) and (7), with (6) ensuring that all vessels arrive at the sink. The vessel capacity is modeled in Constraints (8) (for reefer containers) and (9) (for all containers), and these are only enforced if a vessel uses an arc. Note that Constraints (8) do not take into account empty reefer equipment, since empty containers do not need power, and can therefore be placed anywhere on the vessel. Cargo is only allowed to flow on arcs where a vessel is sailing 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.

Flexible arcs have a duration that is constrained by the minimum and maximum sailing time of the vessel on the arc in Constraints (13). The enter and exit time of a vessel at inflexible ports is handled by Constraints (14) and (15). In practice, these constraints are only necessary if one of the outgoing arcs from an inflexible visit ends at a flexible visit. Constraints (16) set the enter time of a visit to be the duration of a vessel on a flexible arc plus the exit time of the vessel at the start of the arc. Constraints (17) set the amount of time a vessel spends at a flexible visit. The first part of the constraint computes the time required to load and unload cargo and equipment, with the final term of the constraint adding the piloting time to the duration only if one of the incoming arcs is enabled (i.e., the flexible visit is being used).

This mathematical model has been shown in the literature to be effective for solving small LSFRP instances (less than 4 or 5 ships), but runs out of time or memory on larger instances [17]. To support decision makers in industry, heuristics are needed that can solve even large LSFRP instances in little time.

3 Related Work

Despite the relevance of the LSFRP to daily liner carrier operations, it was not mentioned in the most recent survey of work in the liner shipping domain [4]. Work in liner shipping optimization has mainly focused on strategic problems such as network design (e.g., [3]) and fleet deployment (e.g., [11]), as well as tactical problems such as cargo allocation/routing [6]. The LSFRP is different due to its relatively short operational timespan and detailed view of shipping operations. However, a similarity to the literature on the LSFRP and these other liner shipping optimization problems is that they have all been solved using only a single objective function.

As identified in a review of multi-objective DSSs in maritime transportation by Mansouri, Lee and Aluko [8], very few papers in maritime transportation consider multiple objectives, other than as components of a single, monetary-based objective function. In fact, the only articles we found solving an operational liner shipping problem with multi-objective techniques consider empty container repositioning with two objectives in [19,20]. The total cost of repositioning the containers within a network is balanced with the amount of unsatisfied demand.

Several DSSs exist within maritime transportation for assisting companies. The TurboRouter system [5] supports fleet scheduling planners, but does not explicitly provide a way of viewing a tradeoff between objectives. A DSS designed specifically for the LSFRP [10] also does not explicitly consider multi-objective optimization, but could be easily augmented with our proposed approach in this work. The StowMan[s] system [7] assists in the stowage of a container vessel. The system does not explicitly generate a Pareto front, but it does allow users to interact with objective weightings and hard constraints, as well as compare the solutions found with various performance indicators.

Models in the liner shipping literature often assign a dollar value to all portions of the objective function. In many cases this makes sense. In the worst case, however, the dollar values given are difficult to compute or even arbitrary, for example, the value of repositioning an empty container in an operational liner shipping network. We suspect the dearth of literature in this area is due to the relative youth of the area of maritime optimization in comparison to other problem domains and large scale of the optimization problems found. In this work, we report good news, as exploring the Pareto front of the LSFRP turns out to not be significantly more difficult than optimizing a weighted objective function.

4 Multi-objective Simulated Annealing Approach

We propose a multi-objective simulated annealing algorithm (MOSA) for solving the LSFRP, based heavily on the existing simulated annealing algorithm in [12]. The intuition behind the algorithm is that we simply conduct a standard SA with reheating using a weighted objective function, saving solutions along the way using an archiving strategy. This works surprisingly well at finding a variety of solutions to the LSFRP. We first describe the solution representation used for the MOSA, the local search neighborhoods used, then the objective functions we consider, and finally we formalize and discuss the MOSA method for the LSFRP.

4.1 Solution Representation

The solution to the LSFRP can be represented as a sequence of visits for each vessel starting from the vessel's initial visit and ending at the graph sink, which is connected to the final phase-in visit of each slot of the goal service. A key requirement of the LSFRP is that the vessel paths are *node disjoint*, as in most ports there is neither the space at the quay nor the necessity of having multiple repositioning vessels at the same place at the same time. However, we relax this constraint in the solution representation and allow vessels to have the same node on their paths and introduce a penalty function to discourage such solutions. Additionally, we allow and penalize infeasible sailing times to flexible visits.

4.2 Neighborhoods

We use three "basic" neighborhoods and two compound neighborhoods within the MOSA. Our basic neighborhoods add, remove or swap visits between the paths of the vessels, while the compound neighborhoods reconstruct large portions of the paths. These neighborhoods were computationally investigated in [12].

Visit Addition. A ship is selected uniformly at random along with an arc (u, v) on its path. A new visit is chosen such that the sailing from u to the new visit and back to v is feasible (in the sense of the penalized objective function). If no such visit exists, the neighborhood simply makes no changes to the solution.

Visit Removal. A ship is selected uniformly at random as is a visit on its path. This visit may not be the first or last visit of the path. The visit is removed from the path if there is an arc that can be added from the previous node to the successor node of the removed node. Otherwise, nothing is changed.

Visit Swap. Two ships are chosen uniformly at random and a visit is chosen from the path of one of the ships. If there is a visit on the path of the other ship, such that swapping the visits between the paths is possible, then these are swapped. Otherwise the solution is not changed.

Random Path Completion (RPC). A ship is selected uniformly at random along with a visit on its path. All visits subsequent to this visit are removed from its path, and a random path to the graph sink is generated. We avoid loops by requiring that no visit appears in the vessel's path more than once. If it is impossible to complete a random path without a loop, the neighborhood is abandoned and the solution is not changed, however we never encountered this situation on real data.

Demand Destination Completion (DDC). This neighborhood attempts to make a small change to a vessel path to increase the amount of containers carried. A vessel is selected at random along with a visit on the vessel's path. A demand is chosen that could be loaded at the visit, but is not carried because the destination of the demand is not on the vessel's path. Using a breadth first search, we attempt to find a path from any visit on the path before the selected visit to the destination of the demand. If one is found, we then search for a path back to a successive visit on the path in the same way. All nodes between the chosen visit and where the path is rejoined are dropped from the path.

Additional Neighborhoods. We note that in [1] a reactive tabu search for the LSFRP is proposed that contains two additional neighborhoods to those that we propose here. These neighborhoods can be seamlessly integrated into our MO-LSFRP approach, and could potentially find even better solutions.

4.3 Objective Functions

We consider three objective functions in this work that in previous work were converted into dollar amounts and summed together (see (1), (2) and (3) in Sect. 2.1.).

Sailing Cost (Minimize). This objective function sums the cost of sailing for a vessel on all of its arcs, both those with fixed sailing times and those where the sailing duration must be determined during the optimization. Only fuel costs and port fees are considered here; ship leasing and crew/hotel costs are ignored, as they are fixed over the timespan of the repositioning, regardless of the route taken. We note that low values of this objective function correspond to less CO_2 and SO_x emissions, and larger values may be associated with higher eco-efficiency depending on the amount of containers transported on the ships.

Cargo Profit (Maximize). The first objective is the cargo profit from transporting customers' containers. This function is made up of the revenue earned per TEU transferred from one port to another minus the cost of loading and unloading the containers from the vessel. Note that transshipment is not necessary in this model.

Equipment Profit (Maximize). This objective counts the number of empty containers transported between ports. This objective function is particularly hard to convert into a dollar amount, as the value of having an empty container is not easily quantified. In cases where no empty container is available in a place where one is needed, carriers might be able to lease a container at market rates. However, these rates vary significantly. Furthermore, carrying empty containers on a repositioning saves money a carrier may have otherwise spent moving the empty containers either on other ships in the network or on extra ships that are chartered just to carry empty containers.

4.4 MOSA

Algorithm 1 extends the SA algorithm in [12] with our MOSA approach. The algorithm accepts the following parameters. The set of functions f represents the three objective functions, while f' is the weighted function with arbitrary weights defined by the algorithm user. The initial temperature of the SA is given by t^{Init}, the cooling factor by α and the reheating factor with β. A single iteration of the SA is assumed to have converged if the temperature falls below t^{Min}, and r^{Itrs} gives the maximum number of iterations without an improvement before convergence is assumed. After $r^{Restart}$ reheats, the search is restarted, but converges after a maximum of r^{Reheat} SA reheats without any improvement.

The algorithm starts by creating an initial solution through one of three construction heuristics proposed in [12]. We then begin the main reheat loop, followed by the SA loop. On line 11, one of the neighborhoods described above is

Algorithm 1. The MOSA algorithm with reheating and restarts.

1: **function** SA(f, f', t^{Init}, α, β, t^{Min}, r^{Itrs}, $r^{Restart}$, r^{Reheat})
2: $s^{Init} \leftarrow$ CreateSolution(); $s^* \leftarrow s^{Init}$; $s^*_{prev} \leftarrow s^{Init}$
3: $t \leftarrow t^{Init}$; $reheats \leftarrow 0$; $nonImprovingReheats \leftarrow 0$
4: $A \leftarrow \emptyset$ ▷ Solution archive
5: **repeat** ▷ Reheat/restart loop.
6: $nonImprovingItr \leftarrow 0$;
7: **if** $reheats \geq r^{Restart}$ **then**
8: $s \leftarrow s^{Init}$
9: $reheats \leftarrow 0$
10: **repeat** ▷ SA loop.
11: $s' \leftarrow$ SelectNeighbor(s)
12: **if** $f(s') \succ f(s)$ **then**
13: UpdateArchive(A, s')
14: $s \leftarrow s'$
15: **else**
16: **if** $\exp(\frac{f'(s')-f'(s)}{t}) >$ Random$(0,1)$ **then** $s \leftarrow s'$ ▷ Metropolis criterion.
17: $nonImprovingItr \leftarrow nonImprovingItr + 1$
18: **if** $f'(s') > f'(s^*)$ **then** $s^* \leftarrow s'$
19: $t \leftarrow t\alpha$
20: **until** $t < t^{Min}$ **or** $nonImprovingItr \geq r^{Itrs}$
21: $t \leftarrow t^{Init}\beta$ ▷ Reheat to a factor β of the initial temperature.
22: $reheats \leftarrow reheats + 1$
23: **if** $s^* \leq s^*_{prev}$ **then** $nonImprovingReheats \leftarrow nonImprovingReheats + 1$
24: $s^*_{prev} \leftarrow \max\{s^*, s^*_{prev}\}$
25: **until** *Time limit reached* **or** $nonImprovingReheats \geq r^{Reheat}$
26: **return** (A, s^*)

chosen uniformly at random and applied to the current candidate solution s. If the neighboring solution s' Pareto dominates the current solution s, it is added to the solution archive (UpdateArchive) and the candidate is updated. We define the domination operator $\succ$ as follows. We save solutions that are better than the solutions in the archive in at least one objective function value. Note that f includes the penalty function as an objective, however any solution with no penalty violations dominates all solutions with penalty violations, meaning that as soon as a solution is found that is feasible, the archive is purged of infeasible solutions. This ensures that MOSA remains an anytime algorithm if a planner has to stop it before it has found feasible solutions, but in the case that feasible solutions are available (as is the case for every instance we have encountered), these are automatically preferred to any infeasible ones.

When no dominating solution is observed, the metropolis acceptance criterion is consulted to determine whether to accept the neighboring solution. Afterwards, the current best solution according to the weighted objective, s^*, is updated. Although not strictly necessary, we keep track of s^* because this solution may be of particular interest to the repositioning coordinator.

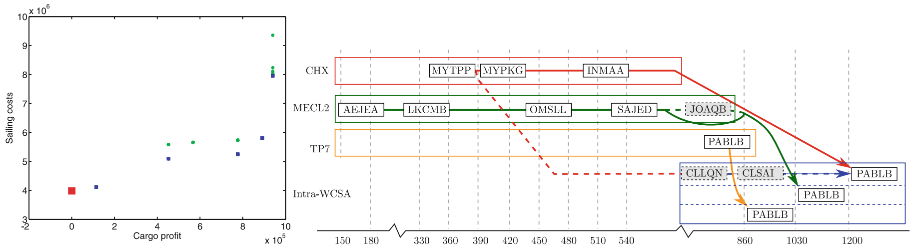
If a reheat is necessary, the temperature is reset on line 21, and the number of non improving reheats is updated. The algorithm continues until a maximum time is reached, or the number of non-improving reheats is exceeded. The solution archive containing the Pareto front, along with s^*, is returned.

5 Experimental Evaluation

We test our multi-objective approach experimentally on the dataset of public LSFRP instances from [12]. Several of these instances are anonymized versions of real repositioning scenarios from a large liner carrier, with the remaining instances being crafted scenarios using real (anonymized) data. We conduct our experiments on a PC with an Intel i7-2600K CPU at 3.4 GHz running Ubuntu 14.04. We allow up to 8 GB of RAM per execution of our approach, but note that our heuristic is not memory bound. For each of the 44 instances in the dataset, we run MOSA 25 times for 60 s each and record the solutions in the Pareto front each time. In this section, we show representations of the Pareto front for two-dimensional cases. For more objectives we apply so-called level diagrams, proposed by [2]. Level diagrams allow for analysis and decision making considering n-dimensional Pareto fronts. The infinity norm over all normalized objective values for a given solution is plotted against the original objective value under consideration. This implicitly leads to a v-shaped plot where points are placed according to their proximity to an ideal point. A decision maker can then determine a level of tolerated loss according to the normalized objectives (up from the peak of the v-shape), and see the possible range of values for a given objective.

Case Study 1: Intra-WCSA. We first investigate the optimization of a small instance, repos2p, in which three ships must be sent to the Intra-WCSA service shown previously in Fig. 1. Figure 2a shows the Pareto front across all runs of the instance. Although this is a small instance with only 36 graph nodes, 150 arcs, and 28 demands, the planner has a number of options he or she can carry out. The solution with the lowest sailing costs (i.e., the lowest fuel consumption) also has the lowest amount of cargo profit. This is not so surprising, as fuel savings come through visiting fewer ports. Indeed, two out of three vessels sail directly from their phase-out port to the phase-in, and the third vessel utilizes the Suez canal (followed by the panama canal) to reach the goal service.

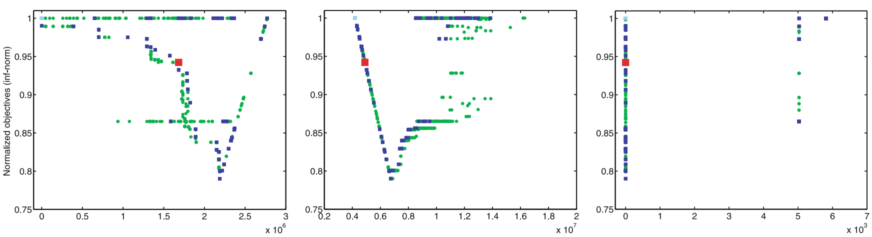
At the opposite end of the solution spectrum, the highest profit generating solution (on the Pareto front) is also associated with the highest sailing costs. In this solution, one vessel sails directly to its phase-in, while the other two vessels go out of their way to deliver cargo in both the Middle East and the east coast of the United States. The two solutions on the Pareto front with roughly 5e6 in sailing costs show how taking a slightly different route can vastly change the cargo profit. These are visualized in Fig. 2b, with dashed lines showing the changes necessary to move from the low profit solution to the high profit solution. The boxes provide the "UN Locode" of each visited port, consisting of a two letter country code and a 3 character port identifier. Adding a visit to JOAQB to the path of the green vessel (with a slightly different goal service start for one vessel) results in the solution with profit 8e5. If for some reason the planner cannot send a ship to JOAQB (no space at the quay, etc.), an alternative is available by extending the phase-out on CHX to INMAA before leaving for PABLB.



(a) Pareto front (b) Solution visualization

Fig. 2. Pareto front computed over all 25 runs for repos2p, with the meta-Pareto front shown as blue squares, the best weighted solution as a large, red square (weights set as described in Sect. 2.1), and (meta)-dominated solutions as green dots. No empty equipment is present in this instance, and the x-axis is measured in USD. The figure on the right shows the repositioning at point (4e5, 5e6) with the x-axis representing the time in hours. (Color figure online)

Case Study 2: SAECS. Our second case study involves instance repos29p, which repositions seven Panamax sized ships, with three starting on an Asia-Europe service (AE), three starting on a Middle East to Europe service (ME), and one starting on a transatlantic route (TA). The goal service is a Europe-Africa line called SAECS that connects northern Europe, Spain and South Africa. The repositioning coordinator is presented with a rich variety of options for solving this instance. Even on only just one execution of MOSA, we see several potential paths for every vessel involved in the repositioning, including at least one "interesting" path that does more than just sail from the phase out to the phase in. The level diagrams in Fig. 3 show how the solutions found compare for the three objectives. Figure 3(c) shows that there are only three different options for equipment: either none is taken or equipment is taken at the levels of 5e3 and under 6e3.



(a) Cargo profit (b) Sailing costs (c) Equipment

Fig. 3. Level diagrams computed over all 25 runs for repos29p, with the meta-Pareto front shown as blue squares, the best weighted solution as a large, red square (weights set as described in Sect. 2.1), and the ten solutions with the lowest sailing cost as light blue squares. (Color figure online)

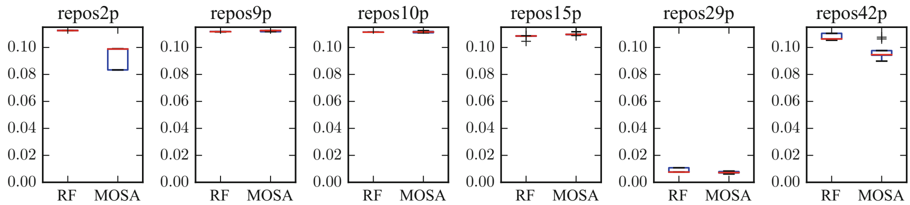


Fig. 4. Comparison of the (normalized) hypervolume of the solutions found by MOSA to a reference front on selected instances.

Comparison of MOSA to a Reference Front. An alternative to generating the Pareto front with MOSA in a single run is to run MOSA several times with different objective weights and store the best solution found in each run. A reference front can then be assembled out of the solutions found. We perform this experiment to see how well MOSA performs against this reference front. The weights we choose for MOSA are 1, 150 and 1 for the cargo, equipment and sailing objectives, respectively, resulting in 3^3 weight configurations. These weights correspond to the values used in the literature for the objective function and were suggested by an industrial partner [17]. We repeat this analysis 10 times, creating 10 reference fronts and compare this against 10 executions of MOSA. We note that computing a reference front is 27 times as expensive as a single execution of MOSA, nonetheless the resulting MOSA fronts are comparable in terms of the amount of hypervolume given the extensive saving in computation time. Figure 4 shows box plots of the normalized hypervolume on several instances for the reference front (RF) and MOSA. We further note that the size of the front is roughly 3.9 times larger using MOSA than the reference front, meaning decision makers are provided with a wider range of non-dominated options.

6 Conclusion and Future Work

This work presents a multiobjective extension of the liner shipping fleet repositioning problem. The problem involves a number of constraints as well as three objectives. We show clear benefits from presenting multiple solutions that balance the fulfillment of the (single-objective) optimization. Furthermore, we avoid requiring decision makers to specify arbitrary weight values for the objectives, while nonetheless ensuring that comparisons between alternative solutions can be made. We also show that level diagrams greatly facilitate the analysis of the generated solutions.

In future work we will extend the embedding of the multiobjective approach within a single-objective optimizer by designing a specialized multi-objective strategy and possibly include further objective functions.

Acknowledgements. Christian Grimme and Heike Trautmann acknowledge support from the European Center for Information Systems (ERCIS).

References

1. Becker, M., Tierney, K.: A hybrid reactive tabu search for liner shipping fleet repositioning. In: Corman, F., Voß, S., Negenborn, R.R. (eds.) ICCL 2015. LNCS, vol. 9335, pp. 123–138. Springer, Cham (2015). doi:10.1007/978-3-319-24264-4_9
2. Blasco, X., Herrero, J., Sanchis, J., Martínez, M.: A new graphical visualization of n-dimensional pareto front for decision-making in multiobjective optimization. Inf. Sci. **178**(20), 3908–3924 (2008)
3. Brouer, B., Alvarez, J., Plum, C., Pisinger, D., Sigurd, M.: A base integer programming model and benchmark suite for liner-shipping network design. Transp. Sci. **48**(2), 281–312 (2013)
4. Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D.: Ship routing and scheduling in the new millennium. Eur. J. Oper. Res. **228**(3), 467–483 (2013)
5. Fagerholt, K.: A computer-based decision support system for vessel fleet schedulingexperience and future research. Decis. Support Syst. **37**(1), 35–47 (2004)
6. Guericke, S., Tierney, K.: Liner shipping cargo allocation with service levels and speed optimization. Transp. Res. Part E Logistics Transp. Rev. **84**, 40–60 (2015)
7. INTERSCHALT Maritime Systems AG: StowMan[s]: Efficient stowage planning for higher cargo intake. A case study, November 2015. http://www.interschalt.com/fileadmin/user_upload/StowManS_Case_Study.pdf. Accessed 26 Sep 2016
8. Mansouri, S.A., Lee, H., Aluko, O.: Multi-objective decision support to enhance environmental sustainability in maritime shipping: a review and future directions. Transp. Res. Part E Logistics Transp. Rev. **78**, 3–18 (2015)
9. Meyer, J., Stahlbock, R., Voß, S.: Slow steaming in container shipping. In: 45th Hawaii International Conference on System Science (HICSS 2012), pp. 1306–1314. IEEE (2012)
10. Müller, D., Tierney, K.: Decision support and data visualization for liner shipping fleet repositioning. Inf. Technol. Manage. 1–19 (2016)
11. Perakis, A., Jaramillo, D.: Fleet deployment optimization for liner shipping Part 1. Background, problem formulation and solution approaches. Marit. Policy Manage. **18**(3), 183–200 (1991)
12. Tierney, K., Askelsdóttir, B., Jensen, R., Pisinger, D.: Solving the liner shipping fleet repositioning problem with cargo flows. Transp. Sci. **49**(3), 652–674 (2014)
13. Tierney, K., Coles, A., Coles, A., Kroer, C., Britt, A., Jensen, R.: Automated planning for liner shipping fleet repositioning. In: McCluskey, L., Williams, B., Silva, J., Bonet, B. (eds.) Proceedings of the 22nd International Conference on Automated Planning and Scheduling, pp. 279–287 (2012)
14. Tierney, K., Jensen, R.M.: The liner shipping fleet repositioning problem with cargo flows. In: Hu, H., Shi, X., Stahlbock, R., Voß, S. (eds.) ICCL 2012. LNCS, vol. 7555, pp. 1–16. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33587-7_1
15. Tierney, K., Jensen, R.M.: A node flow model for the inflexible visitation liner shipping fleet repositioning problem with cargo flows. In: Pacino, D., Voß, S., Jensen, R.M. (eds.) ICCL 2013. LNCS, vol. 8197, pp. 18–34. Springer, Heidelberg (2013). doi:10.1007/978-3-642-41019-2_2
16. Tierney, K., Pacino, D., Jensen, R.: On the complexity of container stowage planning problems. Discrete Appl. Math. **169**, 225–230 (2014)
17. Tierney, K.: Optimizing Liner Shipping Fleet Repositioning Plans. Springer, Cham (2015)
18. United Nations Conference on Trade and Development (UNCTAD): Review of maritime transport (2015)

19. Wong, E., Lau, H., Mak, K.: Immunity-based evolutionary algorithm for optimal global container repositioning in liner shipping. OR Spectr. **32**(3), 739–763 (2010)
20. Wong, E., Yeung, H., Lau, H.: Immunity-based hybrid evolutionary algorithm for multi-objective optimization in global container repositioning. Eng. Appl. Artif. Intell. **22**(6), 842–854 (2009)