

# Big Data Optimization in Maritime Logistics

Berit Dangaard Brouer, Christian Vad Karsten and David Pisinger

**Abstract** Seaborne trade constitutes nearly 80 % of the world trade by volume and is linked into almost every international supply chain. Efficient and competitive logistic solutions obtained through advanced planning will not only benefit the shipping companies, but will trickle down the supply chain to producers and consumers alike. Large scale maritime problems are found particularly within liner shipping due to the vast size of the network that global carriers operate. This chapter will introduce a selection of large scale planning problems within the liner shipping industry. We will focus on the solution techniques applied and show how strategic, tactical and operational problems can be addressed. We will discuss how large scale optimization methods can utilize special problem structures such as separable/independent sub-problems and give examples of advanced heuristics using divide-and-conquer paradigms, decomposition and mathematical programming within a large scale search framework. We conclude the chapter by discussing future challenges of large scale optimization within maritime shipping and the integration of predictive big data analysis combined with prescriptive optimization techniques.

**Keywords** Large-scale optimization · Decision support tools · Prescriptive analytics · Maritime logistics

## 1 Introduction

Modern container vessels can handle up to 20,000 twenty-foot equivalent units (TEU) as seen on Fig. 1. The leading companies may operate a fleet of more than 500 vessels and transport more than 10,000,000 full containers annually that need

---

B.D. Brouer · C.V. Karsten · D. Pisinger (✉)

Department of Management Engineering, Technical University of Denmark,  
Produktionstorvet, Building 426, 2800 Kgs Lyngby, Denmark  
e-mail: dapi@dtu.dk

B.D. Brouer  
e-mail: berit@brouer.com

C.V. Karsten  
e-mail: cvadkarsten@gmail.com



**Fig. 1** Seaborne trade constitutes nearly 80 % of the world trade by volume, and calls for the solution of several large scale optimization problems involving big data. *Picture: Maersk Line*

to be scheduled through the network. There is a huge pressure to fill this capacity and utilize the efficiency benefits of the larger vessels but at the same time markets are volatile leading to ever changing conditions. Operating a liner shipping network is truly a big-data problem, demanding advanced decisions based on state-of-the art solution techniques. The digital footprint from all levels in the supply chain provides opportunities to use data that drive a new generation of faster, safer, cleaner, and more agile means of transportation. Efficient and competitive logistic solutions obtained through advanced planning will not only benefit the shipping companies, but will trickle down the supply chain to producers and consumers.

Maritime logistics companies encounter large scale planning problems at both the strategic, tactical, and operational level. These problems are usually treated separately due to complexity and practical considerations, but as will be seen in this chapter the decisions are not always independent and should not be treated as such. Large scale maritime problems are found both within transportation of bulk cargo, liquefied gasses and particularly within liner shipping due to the vast size of the network that global carriers operate. In 2014 the busiest container terminal in the world, Port of Shanghai, had a throughput of more than 35,000,000 TEU according to Seatrade Global, which is also approximately the estimated number of containers in circulation globally. This chapter will focus on the planning problems faced by a global carrier operating a network of container vessels and show how decision support tools based on mathematical optimization techniques can guide the process of adapting a network to the current market.

At the strategic level carriers determine their fleet size and mix along with which markets to serve thus deciding the layout of their network. The network spanning the globe serving tens of thousands of customers leads to a gazillion possible configurations for operating a particular network. At the tactical level schedules for the individual services and the corresponding fleet deployment is determined, while the routing of containers through the physical transportation network, stowage of containers on the vessels, berthing of the vessels in ports, and disruption management due to e.g. bad weather or port delays is handled at the operational level. In general

these problems can be treated separately, but as the layout of the network will affect e.g. the routing of the containers the problems are far from independent.

Operational data can lead to better predictions of what will happen in the future and carriers are constantly receiving sensor data from vessels that can help predict e.g. disruptions or required maintenance and similarly, data received from terminals can be used to predict delays and help vessels adjust sailing speed to save fuel. But given a predicted future scenario it may still not be obvious what the best actions are neither at the strategic, tactical or operational level. A large shipping company may be capable of producing good estimates of future demand and oil price fluctuations, or predicting possible disruptions. Under certain circumstances these predictions may require simple independent actions to adjust the network, but it is more likely that the actions will be dependent on other factors in the network. In that case difficult and complex here-and-now decisions must be made to adjust the transportation network optimally to the new situation. When there is a large number of decisions to be made and when the decisions influence each other prescriptive models based on optimization can help make the best choice. Predictive and prescriptive methods combined can serve as decision support tools and help select the best strategy, where the predictions made by machine learning algorithms, can be fed into large scale optimization algorithms to guide the decision process faced by carriers.

Most data in liner shipping are associated with some degree of uncertainty. First of all, demands are fluctuating over the year, and even if customers have booked a time slot for their containers these data are affected by significant uncertainty. In liner shipping no fees are paid if the customer is not delivering the booked number of containers, so customers may at any time choose to use another shipping company, or to postpone the delivery. This stimulates overbooking which adds uncertainty to the models. Port availabilities are also highly uncertain. If a vessel sticks to the normal time table, it can generally be assumed that the time slot is available, but if a vessel is delayed or the company wants to change the route, all port calls must be negotiated with the port authorities. This substantially complicates planning, and makes it necessary to use a trial and force method to find a good solution.

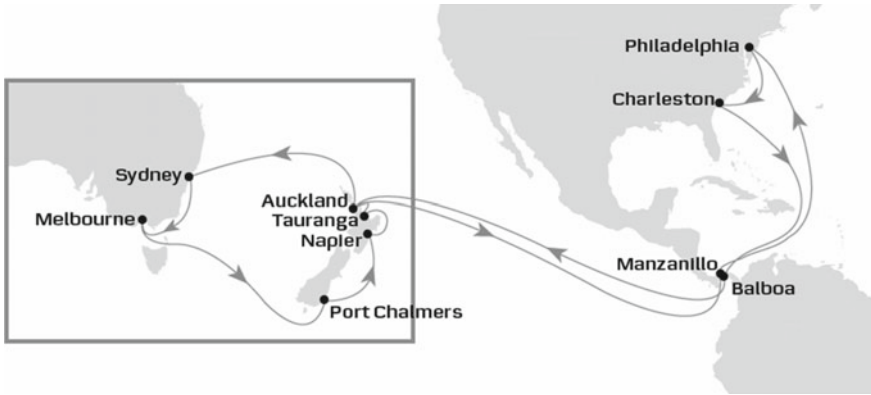
There are several different approaches for solving large scale optimization problems. If a problem exhibit a special separable structure it can be decomposed and solved more efficiently by using either column generation if the complication involves the number of variables or row generation if the number of constraints is too large [5, 8, 18, 20], by dynamic programming [17], or constraint programming [36]. For less structured or extremely large problems it can be advantageous to use (meta)-heuristics to obtain solutions quickly, but often of unknown quality [15, 22]. Finally it is frequently possible, with a good modeling of a problem, to rely solely on Linear Programming, LP, or Mixed Integer Programming, MIP, solvers, see e.g. [42] for a discussion of modeling techniques and the trade-off between stronger versus smaller models. Algorithmic and hardware improvements have over the last three decades resulted in an estimated speed-up for commercial MIP solvers of a 200 billion factor [7], making it feasible not only to solve large linear models but also more

advanced integer decision models of realistic size. In practice a combination of the different techniques is often seen and maritime logistics gives an illustrative case of the importance of all of these large scale optimization methods.

## 2 Liner Shipping Network Design

The Liner Shipping Network Design Problem, LSN DP, is a core planning problem facing carriers. Given an estimate of the demands to be transported and a set of possible ports to serve, a carrier wants to design routes for its fleet of vessels and select which demands of containers to satisfy. A route, or *service*, is a set of similarly sized vessels sailing on a non-simple cyclic itinerary of ports according to a fixed, usually weekly, schedule. Hence the round trip duration for a vessel is assumed to be a multiple of a week and to ensure weekly frequency in the serviced ports a sufficient number of vessels is assigned. If a round trip of the vessel takes e.g. 6 weeks, then 6 vessels are deployed on the same route. To make schedules more robust buffer time is included to account for delays. However, delays may still lead to local speed increases which increases the overall energy consumption. An example of a service can be seen in Fig. 2 which shows the Oceania-Americas Service with a round trip time of 10 weeks. The weekly departures may in some cases simplify the mathematical formulation of the problem, since customer demands and vessel capacities follow a weekly cycle. *Trunk services* serve central main ports and can be both inter and intra regional whereas *feeder services* serve a distinct market and typically visit one single main port and several smaller ports. When the network has been determined the containers can be routed according to a fixed schedule with a predetermined trip duration. A given demand is loaded on to a service at its departure port, which may bring the demand directly to the destination port or the container can be unloaded at one or several intermediate ports for *transshipment* to another service before finally reaching its final destination. Therefore, the design of the set of services is complex, as they interact through transshipments and the majority of containers are transshipped at least once during transport. A carrier aims for a network with high utilization, a low number of transshipments, and competitive transit times. Services are divided into a head- and a back-haul direction. The head haul direction is the most cargo intensive and vessels are almost full. Hence, the head haul generates the majority of the revenue and due to customer demand for fast delivery the head haul operates at increased speeds with nearly no buffer time for delays. The back haul operates at slower speeds with additional buffer time assigned. A delay incurred on the head haul is often recovered during the back-haul.

In practice a carrier will never re-design a network from scratch as there are significant costs associated with the reconfiguration [40]. Rather, the planners or network design algorithms will take the existing network and suggest incremental changes to adjust the network to the current economic environment. Most network changes requires evaluation of the full cargo routing problem to evaluate the quality



**Fig. 2** The Oceania-Americas Service (OC1) from the 2014 Maersk Line Network. *Picture: Maersk Line*

of the network since regional changes can have unintended consequences in the entire network.

Routing of both vessels and containers are in most state-of-the-art methods considered simultaneously [1, 2, 11, 12, 35, 38], as these problems are completely interrelated. However, several of the before mentioned approaches exploit the fact that the problems are separable into two tiers and design algorithms utilizing this structure. The cargo routing reduces to a multicommodity flow problem, MCF, and serves as the lower tier where the revenue of the network is determined. The vessel routing problem reduces to a (more complex) problem of cycle generation and corresponds to the upper tier, where the cost of the network is determined. The following section gives insight to the container routing problem and its relation to the multicommodity flow problem.

### 2.1 Container Routing

We define  $G = (N, A)$  to be a directed graph with nodes  $N$  and edges  $A$ . The node set  $N$  represents the geographical locations in the model i.e. ports and the arc set  $A$  connects the ports. The arcs are determined by the scheduled itineraries and the cargo capacity is determined by the assignment of vessels to the schedule. Let  $K$  be the set of commodities to transport,  $q_k$  be the amount of commodity  $k \in K$  that is available for transport, and  $u_{ij}$  be the capacity of edge  $(i, j)$ . We assume that each commodity has a single origin node,  $O_k$ , and a single destination node,  $D_k$ .

There are two commonly used formulations of the MCF based on either arc or path flow variables. The arc flow formulation can be stated as follows. For each node  $i \in N$  and commodity  $k \in K$  we define  $q(i, k) = q^k$  if  $i = O_k$ ,  $q(i, k) = -q^k$  if  $i = D_k$ , and

$q(i, k) = 0$  otherwise. For each node  $i \in N$  we define the set of edges with tail in node  $i$  as  $\delta^+(i) = \{(j, j') \in A : j = i\}$  and head in node  $i$  as  $\delta^-(i) = \{(j, j') \in A : j' = i\}$ .

With this notation the MCF problem can be stated as the following LP

$$\min \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k \tag{1}$$

$$\text{s.t.} \quad \sum_{(j,j') \in \delta^+(i)} x_{jj'}^k - \sum_{(j,j') \in \delta^-(i)} x_{jj'}^k = q(i, k) \quad i \in N, k \in K \tag{2}$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij} \quad (i, j) \in A \tag{3}$$

$$x_{ij}^k \geq 0 \quad (i, j) \in A, k \in K \tag{4}$$

The objective function (1) minimizes the cost of the flow. The flow conservation constraint (2) ensures that commodities originates and terminates in the right nodes. The capacity constraint (3) ensures that the capacity of each edge is respected. The formulation has  $|K||A|$  variables and  $|A| + |K||N|$  constraints. The number of variables is hence polynomially bounded, but for large graphs like the ones seen in global liner shipping networks this formulation requires excessive computation time and may even be too large for standard LP-solvers (see e.g. [14]).

The *block-angular structure* of the constraint matrix in the arc-flow formulation can be exploited and by *Dantzig-Wolfe decomposition* it is possible to get a reformulation with a master problem considering paths for all commodities, and a subproblem defining the possible paths for each commodity  $k \in K$ . We note that in general any arc flow can be obtained as a convex combination of path flows. In the path-flow formulation each variable,  $f^p$ , in the model corresponds to a path,  $p$ , through the graph for a specific commodity. The variable states how many units of a specific commodity that is routed through the given path, the cost of each variable is given by the parameter  $c_p$ . Let  $P^k$  be the set of all feasible paths for commodity  $k$ ,  $P^k(a)$  be the set of paths for commodity  $k$  that uses edge  $a$  and  $P(a) = \cup_{k \in K} P^k(a)$  is the set of all paths that use edge  $a$ . The model then becomes:

$$\min \sum_{k \in K} \sum_{p \in P^k} c_p f^p \tag{5}$$

$$\text{s.t.} \quad \sum_{p \in P^k} f^p = q_k \quad k \in K \tag{6}$$

$$\sum_{p \in P(a)} f^p \leq u_{ij} \quad (i, j) \in A \tag{7}$$

$$f^p \geq 0 \quad k \in K, p \in P^k \tag{8}$$

The objective function (5) again minimizes the cost of the flow. Constraint (6) ensures that the demand of each commodity is met and constraint (7) ensures that the capacity limit of each edge is obeyed. The path-flow model has  $|A| + |K|$  constraints, but the number of variables is, in general, growing exponentially with the

size of the graph. However, using column generation the necessary variables can be generated dynamically and in practice the path-flow model can often be solved faster than the arc-flow model for large scale instances of the LSND problem [14].

Column generation operates with a reduced version of the LP (5)–(8), which is called the master problem. The master problem is defined by a reduced set of columns  $Q^k \subseteq P^k$  for each commodity  $k$  such that a feasible solution to the LP (5)–(8) can be found using variables from  $\cup_{k \in K} Q^k$ . Solving this LP gives rise to dual variables  $\pi_k$  and  $\lambda_{ij}$  corresponding to constraint (6) and (7), respectively. For a variable  $j \in \cup_{k \in K} P^k$  we let  $\kappa(j)$  denote the commodity that a variable serves and let  $p(j)$  represent the path corresponding to the variable  $j$ , represented as the set of edges traversed by the path. Then we can calculate the *reduced cost*  $\bar{c}_j$  of each column  $j \in \cup_{k \in K} P^k$  as follows

$$\bar{c}_j = \sum_{(i,j) \in p(j)} (c_{ij}^{\kappa(j)} - \lambda_{ij}) - \pi_{\kappa(j)}.$$

If we can find a variable  $j \in \cup_{k \in K} (P^k \setminus Q^k)$  such that  $\bar{c}_j < 0$  then this variable has the potential to improve the current LP solution and should be added to the master problem, which is resolved to give new dual values. If, on the other hand, we have that  $\bar{c}_j \geq 0$  for all  $j \in \cup_{k \in K} (P^k \setminus Q^k)$  then we know the master problem defined by  $Q^k$  provides the optimal solution to the complete problem (for more details see [24]). In order to find a variable with negative reduced cost or prove that no such variable exists we solve a sub-problem for each commodity. The sub-problem seeks the feasible path for commodity  $k$  with minimum reduced cost given the current dual values. Solving this problem amounts to solving a shortest path problem from source to destination of the commodity with edge costs given by  $c_{ij} - \lambda_{ij}$  and subtracting  $\pi_k$  from this cost in order to get the reduced cost. As will be seen later we can extend the model to reject demands by including additional variables with an appropriate penalty. When solving the shortest path problem additional industry constraints such as number of transshipments, trade policies, or time limits on cargo trip duration can be included. Including such constraints will increase the complexity of the sub-problem as the resulting problem becomes a resource constrained shortest path problem. Karsten et al. [24] has made a tailored algorithm for a cargo routing problem considering lead times and show that it does not necessarily increase the solution time to include transit time constraints, mainly because the size of solution space is reduced. Additionally, Karsten et al. [24] give an overview of graph topologies accounting for transshipment operations when considering transit times.

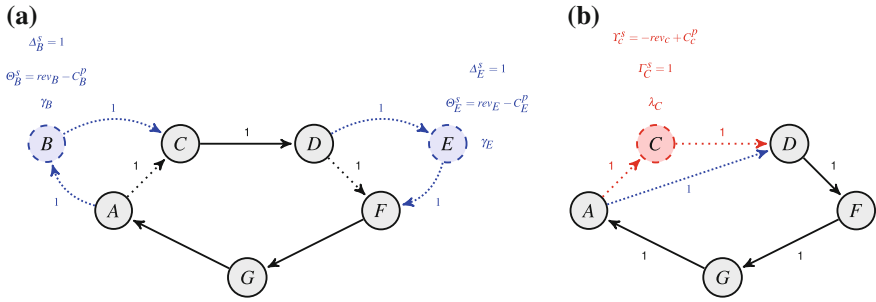
To construct routes used in the upper tier of the network design problem we will go through a more recent approach in the next section which use an advanced mathematical programming based heuristic to solve the problem within a large scale search framework. In general, when a generic network has been designed it is transformed into a physical sailing network by determining a specific schedule, deploying vessels from the available fleet and deciding on the speed and actual flow of containers.

Some aspects of the tactical and operational decisions can of course be integrated in the network design process at the cost of computational tractability, but with the potential benefit of higher quality networks.

### 3 Mat-Heuristic for Liner Shipping Network Design

Mathematical programming models of the LSNDP are closely related to the capacitated fixed charge network design problem [23] in installing a discrete set of capacities for the set of commodities  $K$ . However, the capacity installed must reflect the routing of container vessels according to the specification of a service as defined in the beginning of this section. Therefore, it is also related to pick-up and delivery vehicle routing problems [41], however being significantly harder to solve as a consequence both of the non-simple cyclic routes, the multiple commodities and the vast size of real life networks. As a consequence optimal methods can only solve very insignificant instances of the LSNDP [2, 35] or provide lower bounds [33]. Several algorithms for solving larger instances of the LSNDP can be categorized as *matheuristics* combining mathematical programming with meta heuristics exploiting the two tier structure, where the variables of the upper tier describe a service and variables of the lower tier describe the container routing (for a reference model of the LSNDP see [10]). Agarwal and Ergun [1] apply a heuristic Benders' decomposition algorithm as well as a Branch and Bound algorithm for heuristically generated routing variables, Alvarez [2] applies a tabu search scheme, where the routing variables are generated by a mathematical program based on the dual values of the lower tier MCF problem in each iteration. [10] use a heuristic column generation scheme, where the routing columns are generated by an integer program based on information from both tiers of the LSNDP along with a set of business rules. The integer program in [10] constructs a single, (possibly non-simple) cyclic route for a given service configuration of vessel class and speed. Route construction is based on the Miller-Tucker-Zemlin subtour elimination constraints known from the CVRP to enumerate the port calls in a non-decreasing sequence. This makes high quality routings for smaller instances of the LSNDP, but for large scale instances it becomes necessary to select a small cluster of related ports in order to efficiently solve the integer program used in the heuristic. A different matheuristic approach is seen in [11, 12], where the core component in a large scale neighborhood search is an integer program designed to capture the complex interaction of the cargo allocation between routes. The solution of the integer program provides a set of moves in the composition of port calls and fleet deployment. Meta-heuristics for the LSNDP are challenged by the difficulty of predicting the changes in the multicommodity flow problem for a given move in the solution space without reevaluating the MCF at the lower tier. The approach of [12] relies on estimation functions of changes in the flow and the fleet deployment related to inserting or removing a port call from a given service and network configuration. Flow changes and the resulting change in the revenue are estimated by solving a series of shortest path problems on the residual graph of





**Fig. 3** Illustration of the estimation functions for insertion and removal of port calls. **a** Blue nodes are evaluated for insertion corresponding to variables  $\gamma_i$  for the set of ports in the neighborhood  $N^s$  of service  $s$ . **b** Red nodes are evaluated for removal corresponding to variables  $\lambda_i$  for the set of current port calls  $F^s$  on service  $s$

the current network for relevant commodities to the insertion/removal of a port call along with an estimation of the change in the vessel related cost with the current fleet deployment.

Given a total estimated change in revenue of  $rev_i$  and port call cost of  $C_i^p$  Fig. 3a illustrate estimation functions for the change in revenue ( $\Theta_i^s$ ) and duration ( $\Delta_i^s$ ) increase for inserting port  $i$  into service  $s$  controlled by the binary variable  $\gamma_i$ . The duration controls the number of vessels needed to maintain a weekly frequency of service. Figure 3b illustrate the estimation functions for the change in revenue ( $\Upsilon_i^s$ ) and decrease in duration ( $\Gamma_i^s$ ) for removing port  $i$  from service  $s$  controlled by the binary variable  $\lambda_i$ . Insertions/removals will affect the duration of the service in question and hence the needed fleet deployment modeled by the integer variable  $\omega_s$  representing the change in the number of vessels deployed. The integer program (9)–(16) expresses the neighborhood of a single service,  $s$ .

$$\max \sum_{i \in N^s} \Theta_i \gamma_i + \sum_{i \in F^s} \Upsilon_i \lambda_i - C_V^{e(s)} \omega_s \quad (9)$$

$$\text{s.t. } T_s + \sum_{i \in N^s} \Delta_i^s \gamma_i - \sum_{i \in F^s} \Gamma_i^s \lambda_i \leq 24 \cdot 7 \cdot (n_s^{e(s)} + \omega_s) \quad (10)$$

$$\omega_s \leq M_{e(s)} \quad (11)$$

$$\sum_{i \in N^s} \gamma_i \leq I_s \quad (12)$$

$$\sum_{i \in F^s} \lambda_i \leq R_s \quad (13)$$

$$\sum_{j \in L_i} \lambda_j \leq |L_i|(1 - \gamma_i) \quad i \in N^s \quad (14)$$

$$\sum_{j \in L_i} \lambda_j \leq |L_i|(1 - \lambda_i) \quad i \in F^s \quad (15)$$

$$\lambda_i \in \{0, 1\}, i \in F^s, \quad \gamma_i \in \{0, 1\}, i \in N^s, \quad \omega_s \in \mathbb{Z}. \quad (16)$$

The objective function (9) accounts for the expected change in revenue of the considered insertions and removals along with the weekly vessel cost  $C_V^{e(s)}$  of the vessel class  $e(s)$  deployed to service  $s$ . Constraint (10) considers the expected change in the duration of the service, where  $T_s$  is the current duration and  $n_s^{e(s)}$  is the number of vessels currently deployed to service  $s$ . The possible addition of vessels is bounded by the number of vessels available  $M_{e(s)}$  of type  $e$  in constraint (11). A limit on the number of insertions/removals respectively are introduced in constraints (12)–(13) to reduce the error of the estimation functions for multiple insertions/removals. The estimation functions also depend on the existing port calls for unloading the commodities introduced by the insertions as well as the ports used for rerouting commodities when removing ports. This is handled by introducing a lockset  $L_i$  for each insertion/removal expressed in constraints (14)–(15). The integer program is solved iteratively for each service in the current network and the resulting set of moves are evaluated for acceptance in a simulated annealing framework. The procedure is an improvement heuristic [3] fine tuning a given network configuration. The algorithm in its entirety constructs an initial network using a simple greedy construction heuristic. The improvement heuristic is applied as a move operator for intensification of the constructed solution. To diversify the solution a perturbation step is performed at every tenth loop through the entire set of services. The perturbation step alters the service composition in the network by removing entire services with low utilization and introducing a set of new services based on the greedy construction heuristic for undeployed vessels. To evaluate the metaheuristic the public benchmark suite, LINER-LIB, for liner shipping network design problems is used.

## 4 Computational Results Using LINER-LIB

LINER-LIB 2012 is a public benchmark suite for the LSNDP presented by [10]. The data instances of the benchmark suite are constructed from real-life data from the largest global liner-shipping company, Maersk Line, along with several industry and public stakeholders. LINER-LIB consists of seven benchmark instances available at <http://www.linerlib.org/> (see [10] for details on the construction of the data instances). Each instance can be used in a low, medium, and high capacity case depending on the fleet of the instance. Table 1 presents some statistics on each instance ranging from smaller networks suitable for optimal methods to large scale instances spanning the globe. Currently published results are available for 6 of the 7 instances, leaving the *WorldLarge* instance unsolved.

LINER-LIB contains data on ports including port call cost, cargo handling cost and draft restrictions, distances between ports considering draft and canal traversal, vessel related data for capacity, cost, speed interval and bunker consumptions, and finally a commodity set with quantities, revenue, and maximal transit time. The commodity data reflects the current imbalance of world trade and the associated differentiated revenue. It is tailored for models of the LSNDP, but may provide useful data for related maritime transportation problems.

**Table 1** The instances of the benchmark suite with indication of the number of ports ( $|P|$ ), the number of origin-destination pairs ( $|K|$ ), the number of vessel classes ( $|E|$ ), the minimum ( $\min v$ ) and maximum number of vessels ( $\max v$ )

Category	Instance and description	$ P $	$ K $	$ E $	$\min v$	$\max v$
Single-hub instances	<b>Baltic</b> Baltic sea, Bremerhaven as hub	12	22	2	5	7
	<b>WAF</b> West Africa, Algeciras as hub	19	38	2	33	51
Multi-hub instance	<b>Mediterranean</b> Mediterranean, Algeciras, Tangier, Gioia Tauro as hubs	39	369	3	15	25
Trade-lane instances	<b>Pacific</b> (Asia-US West)	45	722	4	81	119
	<b>AsiaEurope</b> Europe, Middle East, and Far east regions	111	4000	6	140	212
World instance	<b>Small</b> 47 Main ports worldwide identified by Maersk Line	47	1764	6	209	317
	<b>Large</b> 197 ports worldwide identified by Maersk Line	197	9630	6	401	601

Computational results for LINER-LIB are presented in [10, 12, 33]. Brouer et al. [10] presented the first results for the benchmark suite using the reference model [10] with biweekly frequencies for the feeder vessel classes and weekly frequencies for remaining classes. The heuristic column generation algorithm is used to solve all instances but the Large world instance with promising results. [12] present computational results using the reference model with weekly frequencies for all vessel classes which has a more restricted solution space than [10]. As a consequence the solutions from [12] are feasible for the model used in [10], but not vice-versa. However, the computational results of [12] indicate that the matheuristic using an improvement heuristic based on integer programming scales well for large instances and holds the current best known results for the Pacific, World Small and AsiaEurope instances. [33] present a service flow model for the LSNDP using a commercial MIP solver presenting results for the two Baltic and WAF instances of LINER-LIB. For details on the results the reader is referred to the respective papers. LINER-LIB is currently used by researchers at a handful of different universities worldwide and may provide data for future results on models and algorithms for LSNDP.

## 5 Empty Container Repositioning

In extension of the network design process a liner shipping company must also consider revenue management at a more operational level. Requests for cargo can be rejected if it is not profitable to carry the containers, or if bottlenecks in the network make it infeasible. Moreover, empty containers tend to accumulate at importing

regions due to a significant imbalance in world trade. Therefore, repositioning empty containers to exporting regions impose a large cost on liner shippers, and these costs need to be incorporated in the revenue model. Since larger shipping companies at any time have several millions of containers in circulation, these decisions are extremely complex and require advanced solution methods.

Alvarez [2] presented a study of large scale instances of the liner service network design problem. The cargo allocation problem is solved as a subproblem of the tabu search algorithm solving the network design problem. Meng and Wang [26] study a network design problem selecting among a set of candidate shipping lines while considering the container routing problem along with the repositioning of empty containers. The model is formulated as a minimum cost problem and as [21] the model handle loaded end empty containers simultaneously, however it does not allow load rejection and only seek to minimize the cost of transport. Song and Dong [39] consider a problem of joint cargo routing and empty container repositioning at the operational level accounting for the demurrage and inventory cost of empty containers. Like most other works on empty repositioning it is a cost minimizing problem where load rejection is not allowed.

Brouer et al. [14] present a revenue management model for strategic planning within a liner shipping company. A mathematical model is presented for maximizing the profit of cargo transportation while considering the possible cost of repositioning empty containers.

The booking decision of a liner shipper considering empty container repositioning can be described as a specialized multi-commodity flow problem with inter-balancing constraints to control the flow of empty containers.

Similarly to the pure cargo routing problem we can define a commodity as the tuple  $(O_k, D_k, q_k, r_k)$  representing a demand of  $q_k$  in number of containers from node  $O_k$  to node  $D_k$  with a sales price per unit of  $r_k$ . The unit cost of arc  $(i, j)$  for commodity  $k$  is denoted  $c_{ij}^k$ . The non-negative integer variable  $x_{ij}^k$  is the flow of commodity  $k$  on arc  $(i, j)$ . The capacity of arc  $(i, j)$  is  $u_{ij}$ . To model the empty containers an empty super commodity  $k_e$  is introduced. The flow of the empty super commodity is defined for all  $(i, j) \in A$  as the integer variables  $x_{ij}^{k_e}$ . The unit cost of arc  $(i, j)$  for commodity  $k_e$  is denoted  $c_{ij}^{k_e}$ . The empty super commodity has no flow conservation constraints and appear in the objective with a cost and in the bundled capacity and inter-balancing constraints. For convenience the commodity set is split into the loaded commodities and the empty super commodity: Let  $K_F$  be the set of loaded commodities. Let  $K_e$  be the set of the single empty super commodity. Finally, let  $K = K_F \cup K_e$ . The inter-balancing constraints also introduce a new set of variables representing leased containers at a node. The cost of leasing is modeled in the objective. Let  $c_i^l$  be the cost of leasing a container at port  $i$ , while  $l_i$  is the integer leasing variable at port  $i$ . Demand may be rejected, due to capacity constraints and unprofitability from empty repositioning cost. The slack variable  $\gamma_k$  represents the amount of rejected demand for commodity  $k$ .

## 5.1 Path Flow Formulation

In the following we introduce a path flow model which is an extension of model (5)–(8). Again, let  $p$  be a path connecting  $O_k$  and  $D_k$  and  $P_k$  be the set of all paths belonging to commodity  $k$ . The flow on path  $p$  is denoted by the variable  $f^p$ . The binary coefficient  $a_{ij}^p$  is one if and only if arc  $(i, j)$  is on the path  $p$ . Finally,  $c_p^k = \sum_{(i,j) \in A} a_{ij}^p c_{ij}^k$  is the cost of path  $p$  for commodity  $k$ . The master problem is:

$$\max \sum_{k \in K_F} \sum_{p \in P_k} (r_k - c_p^k) f^p - \sum_{(i,j) \in A} c_{ij}^{k_e} x_{ij}^{k_e} - \sum_{i \in N} c_l^i l^i \quad (17)$$

$$\text{s.t.} \quad \sum_{k \in K_F} \sum_{p \in P_k} a_{ij}^p f^p + x_{ij}^{k_e} \leq u_{ij} \quad (i, j) \in A \quad (18)$$

$$\sum_{p \in P_k} f^p + \gamma_k = q_k \quad k \in K_F \quad (19)$$

$$\sum_{k \in K_F} \sum_{p \in P_k} \sum_{j \in N} (a_{ij}^p - a_{ji}^p) f^p + x_{ij}^{k_e} - x_{ji}^{k_e} - l^i \leq 0 \quad i \in N \quad (20)$$

$$f^p \in \mathbb{Z}_+, p \in P_k, \quad \gamma_k \in \mathbb{Z}_+, k \in K_F \quad x_{ij}^{k_e} \in \mathbb{Z}_+, (i, j) \in A, \quad l^i \in \mathbb{Z}_+, i \in N \quad (21)$$

where the  $x_{ij}^k$  variables can be replaced by  $\sum_{p \in P_k} a_{ij}^p f^p$  for all  $k \in K_F$ . The convexity constraints for the individual subproblems (19) bound the flow between the  $(O_k, D_k)$  pair from above (a maximal flow of  $q_k$  is possible).

Paths are generated on the fly using delayed column generation. Brouer et al. [14] report computational results for eight instances based on real life shipping networks, showing that the delayed column generation algorithm for the path flow model clearly outperforms solving the arc flow model with the CPLEX barrier solver. In order to fairly compare the arc and path flow formulation a basic column generation algorithm is used for the path flow model versus a standard solver for the arc flow model. Instances with up to 234 ports and 293 vessels for 9 periods were solved in less than 35 min with the column generation algorithm. The largest instance solved for 12 periods contains 151 ports and 222 vessels and was solved in less than 75 min.

The algorithm solves instances with up to 16,000 commodities over a twelve month planning period within one hour. Integer solutions are found by simply rounding the LP solution. The model of Erera et al. [21] is solved to integer optimality using standard solvers as opposed to the rounded integer solution presented here. The problem sizes of [14] are significantly larger than those of [21] and the rounded integer solutions lead to a gap of at most 0.01 % from the LP upper bound of the path flow formulation, which is very acceptable, and far below the level of uncertainty in the data. The results of [21] confirm the economic rationale in simultaneously considering loaded and empty containers.

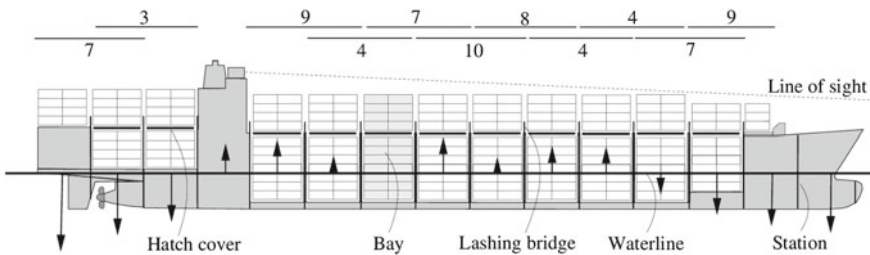
## 6 Container Vessel Stowage Plans

With vessels carrying up to 20,000 TEU, stowage of the containers on board is a non-trivial task demanding fast algorithms as the final load list is known very late. Stowage planning can be split into a master planning problem and a more detailed slot planning problem. The master planning problem should decide a proper mixture of containers, so that constraints on volume, weight, and reefer plugs are respected. The slot planning problem should assign containers to slots in the vessel so that the loading and unloading time in ports can be minimized. The vessel must be seaworthy, meaning that stability and stress constraints must be respected.

Figure 4 illustrates the arrangement of bays in a container vessel. Containers are loaded bottom-up in each bay up to a given stacking height limited by the line of sight and other factors. Some containers are loaded below deck, while other containers are loaded above the hatch cover. The overall weight sum of containers may not exceed a given limit, and the weight need to be balanced. Moreover, torsions should be limited, making it illegal to e.g. only load containers at the same front and end of the vessel. Refrigerated containers (reefers) need to be attached to an electric plug. Only a limited number of plugs are available, and these plugs are at specific positions.

A good stowage plan should make sure that it is not necessary to rearrange containers at each port call. All containers for the given port should be directly accessible when arriving to the port, and there should be sufficient free capacity for loading new containers. If several cranes are available in a port, it is necessary to ensure that all cranes can operate at the same time without blocking for each other.

Pacino [28] presents a MIP model for the master problem. The model is based on Pacino et al. [29, 30]. The model considers both 20' and 40' containers, assuming that two 20' containers can fit in the slot of a 40' container provided that the middle is properly supported. Four types of containers are considered: light, heavy, light reefer, and heavy reefer. Decision variables are introduced for each slot, indicating how many of each container type will be loaded in the slot.



**Fig. 4** The arrangement of bays in a small container vessel, and stacking heights. The *arrows* indicate forces. *Picture:* Pacino [28]

The MIP model has a large number of constraints: First of all, a load list and cargo estimates are used to calculate the number of containers of each type that needs to be stowed. Moreover, every slot has a capacity of dry containers and reefers. An overall weight limit given by the capacity of the vessel is also imposed. When calculating the weight limit, average values for light and heavy containers are used to ease the calculations.

Trim, draft, buoyancy and stability is calculated as a function of displacement and center of gravity of the vessel.

Finally, a number of penalties associated with a given loading are calculated. These include hatch-overstowage, overstowage in slots, time needed for loading, and excess of reefer containers. The objective of the model minimizes a weighted sum of the penalties.

Pacino [28] show that the master planning problem is NP-hard. Computational results are reported for instances with vessel capacity up to around 10,000 TEU, visiting up to 12 ports involving more than 25,000 lifts (crane moves of a container). Several of these instances can be solved within 5 min up to a 5 % gap, using a MIP-solver.

## 6.1 Mathematical Model

In the slot planning phase, the master plan is refined by assigning the containers to specific slots on board the vessel [31]. This problem involves handling of a number of stacking rules, as well as constraints on stack heights and stack weight. Since several of the containers are already stowed on board the vessel the objective is to arrange containers with the same destination port in the same stack, free as many stacks as possible, minimize overstowage, and minimize the number of non-reefer containers assigned to reefer slots. Due to the large number of logical constraints in this problem [19] proposed a logical model using the following notation.  $\mathcal{S}$  is the set of stacks,  $\mathcal{T}_s$  is the set of tiers for stack  $s$ ,  $\mathcal{P}$  represents the aft ( $p = 1$ ) and fore ( $p = 2$ ) of a cell,  $\mathcal{C}$  is the set of containers to stow in the location and  $\mathcal{C}^P \subset \mathcal{C}$  is the subset of containers in the release, i.e. the set of containers that are already on-board the vessel.  $x_{stp} \in \mathcal{C} \cup \{\perp\}$  is a decision variable indicating the location of a container  $c \in \mathcal{C}$  or the empty assignment  $\perp$ .  $A_{stp}^{40}$  is a binary variable indicating if the cell in stack  $s$ , tier  $t$ , and position  $p$  can hold a 40' foot container and similarly  $A_{stp}^{20}$  is one if a slot can hold a 20' container.  $A_{stp}^R$  is a binary indicator for the position of reefer plugs.  $W_s$  and  $H_s$  is the maximum weight and height of stack  $s$ . The attribute functions use  $w(c)$  and  $h(c)$  for the weight and height of a container.  $r(c)$  is true iff the container is a reefer,  $\perp(c)$  is true iff  $c = \perp$ ,  $f(c)$  is true iff the container is 40', and  $t(c)$  is true iff it is a 20' container. Then the logical model is:

$$|\{x_{stp} = c | s \in \mathcal{S}, t \in \mathcal{T}_s, p \in \mathcal{P}\}| = 1 \quad c \in \mathcal{C} \quad (22)$$

$$x_{s_c t_c p_c} = c \quad c \in \mathcal{C}^P \quad (23)$$

$$\neg f(x_{st1}) \wedge f(x_{st2}) \implies \perp(x_{st1}) \quad s \in \mathcal{S}, t \in \mathcal{T}_s \quad (24)$$

$$t(x_{stp}) \implies A_{stp}^{20} \quad s \in \mathcal{S}, t \in \mathcal{T}_s, p \in \mathcal{P} \quad (25)$$

$$f(x_{st1}) \implies A_{st}^{40} \quad s \in \mathcal{S}, t \in \mathcal{T}_s \quad (26)$$

$$\sum_{t \in \mathcal{T}_s} (w(x_{st1}) + w(x_{st2})) \leq W_s \quad s \in \mathcal{S} \quad (27)$$

$$\sum_{t \in \mathcal{T}_s} \max(h(x_{st1}), h(x_{st2})) \leq H_s \quad s \in \mathcal{S} \quad (28)$$

$$\neg \perp(x_{stp}) \implies (t(x_{s(t-1)1}) \wedge t(x_{s(t-1)2})) \vee f(x_{s(t-1)1}) \quad s \in \mathcal{S}, t \in \mathcal{T}_s \setminus \{1\}, p \in \mathcal{P} \quad (29)$$

$$f(x_{st1}) \implies \perp t(x_{s(t+1)p}) \quad s \in \mathcal{S}, t \in \mathcal{T}_s \setminus \{N_s^T\}, p \in \mathcal{P} \quad (30)$$

$$r(x_{stp}) \wedge t(x_{stp}) \implies A_{stp}^R \quad s \in \mathcal{S}, t \in \mathcal{T}_s, p \in \mathcal{P} \quad (31)$$

$$r(x_{st1}) \wedge f(x_{st1}) \implies A_{st1}^R \vee A_{st2}^R \quad s \in \mathcal{S}, t \in \mathcal{T}_s \quad (32)$$

Constraints (22)–(23) ensure that each container is assigned to exactly one slot. Constraint (24) ensures that a 40' container occupies both the aft and fore position of a cell. The assignments need to respect cell capacity (25)–(26), stack height and stack weight limits (27)–(28). Two 20' containers can be stowed in a 40' slot, if properly supported from below (29). This means that 40' container can be stacked on top of two 20' containers, but not the other way around (30). Reefer containers need to be assigned to slots with a power plug (31)–(32).

In order to minimize the objective function [19] propose to use Constraint-Based Local Search. The framework combines local search algorithms with constraint programming. The constraint satisfaction part of the problem is transformed to an optimization problem where the objective is to minimize constraint violation. A hill-climbing method is used to optimize the slot planning. The neighborhood in the search consists of swapping containers between a pair of cells.

Pacino [28] report computational results for 133 real-life instances, showing that the local search algorithm actually finds the optimal solution in 86 % of the cases. The running times are below 1 second.

## 7 Bunker Purchasing

In a liner shipping network bunker fuel constitutes a very large part of the variable operating cost for the vessels. Also, the inventory holding costs of the bunker on board may constitute a significant expense to the liner shipping company.



Bunker prices are fluctuating and generally correlated with the crude oil price, but there are significant price differences between ports. This creates the need for frequent (daily) re-optimization of the bunker plan for a vessel, to ensure the lowest bunker costs.

Bunker can be purchased on the spot market when arriving to a port, but normally it is purchased some weeks ahead of arrival. Long-term contracts between a liner shipping company and a port can result in reduced bunkering costs by committing the company to purchase a given amount of bunker. Bunkering contracts may cover several vessels sailing on different services, making the planning quite complex.

The bunker purchasing problem is to satisfy the vessels consumption by purchasing bunkers at the minimum overall cost, while considering reserve requirements, and other operational constraints. Bunker purchasing problems involve big data. Real-life instances may involve more than 500 vessels, 40,000 port calls, and 750 contracts.

For a vessel sailing on a given port to port voyage at a given speed, the bunker consumption can be fairly accurately predicted. This gives an advantage in bunker purchasing, when a vessel has a stable schedule known for some months ahead. The regularity in the vessel schedules in liner shipping allows for detailed planning of a single vessel.

Besbes and Savin [9] consider different re-fueling policies for liner vessels and present some interesting considerations on the modeling of stochastic bunker prices using Markov processes. This is used to show that the bunkering problem in liner shipping can be seen as a stochastic capacitated inventory management problem. Capacity is the only considered operational constraint. More recently [43] examined re-fueling under a worst-case bunker consumption scenario.

The work of [34] considers multiple tanks in the vessel and stochasticity of both prices and consumption, as well as a range of operational constraints. [44] does not consider stochastic elements nor tanks, but has vessel speed as a variable of the model. The work of [25] minimizes bunker costs as well as startup costs and inventory costs for a single liner shipping vessel. This is done by choosing bunker ports and bunker volumes but also having vessel round trip speed (and thus the number of vessels on the service) as a variable of the model.

In [37] a model is developed which considers the uncertainty of bunker prices and bunker consumption, modeling their uncertainty by markov processes in a scenario tree. The work can be seen as an extension of [44], as it considers vessel speed as a variable within the same time window bounds. Capacity and fixed bunkering costs is considered, as is the holding / tied capital cost of the bunkers.

The studies described above do not consider bunker contracts, and all model the bunker purchasing for a single vessel.

### 7.1 Bunker Purchasing with Contracts

Plum et al. [32] presented a decomposition algorithm for the Bunker Purchasing with Contracts Problem, BPCP, and showed that the model is able to solve even very large real-life instances. The model is based on writing up all bunkering patterns, and hence may be of exponential size. Let  $I$  be the set of ports visited on an itinerary,  $B$  be the set of bunker types, and  $V$  be the set of vessels. A contract  $c \in C$  has a minimal  $\underline{q}_c$  and maximal  $\bar{q}_c$  quantity that needs to be purchased. A contract  $c$  will give rise to a number of purchase options  $m \in M$ , i.e. discrete events where a specific vessel  $v$  calls a port within the time interval of a contract  $c$ , allowing it to purchase bunker at the specific price  $p_m$ . Each time a purchase is done at port  $i$  a startup cost  $sc_i$  is paid.

Let  $R_v$  be the set of all feasible bunkering patterns for a vessel  $v$ . A bunkering pattern is feasible if a sufficient amount of bunker is available for each itinerary, including reserves. Bunker is available in various grades, and it is allowed to substitute a lower grade with a higher grade. In some areas, only low-sulphur bunker may be used, and this needs to be respected by the bunkering plan. Moreover initial and terminal criteria for bunker volumes must be met. Finding a legal bunkering pattern can be formulated as a MIP model [32] and solved by commercial solvers. Each pattern  $r \in R_v$  is denoted as a set of bunkerings.

Let  $u_r = \sum_{m \in M} (p_m l_m) + \sum_{i \in I} \sum_{v \in V} \sum_{b \in B} (\delta_{i,b} sc_i)$  be the cost for pattern  $r \in R_v$ . In this expression,  $l_m$  is the purchase of bunker for each purchase option  $m$ . and  $p_m$  is the price of option  $m$ . The binary variable  $\delta_{i,b}$  is set to one iff a purchase of bunker type  $b$  is made at port call  $i$ . Let  $\lambda_r$  be a binary variable, set to 1 iff the bunkering pattern  $r$  is used. Let  $o_{r,c}$  be the quantity purchased of contract  $c$  by pattern  $r$ . The BPCP can then be formulated as

$$\min \sum_{v \in V} \sum_{r \in R_v} \lambda_r u_r + \sum_{c \in C} (\underline{s}_c \underline{w} + \bar{s}_c \bar{w}) \tag{33}$$

$$\text{s.t. } \underline{q}_c - \underline{s}_c \leq \sum_{v \in V} \sum_{r \in R_v} \lambda_r o_{r,c} \leq \bar{q}_c + \bar{s}_c \quad c \in C \tag{34}$$

$$\sum_{r \in R_v} \lambda_r = 1 \quad v \in V \tag{35}$$

$$\lambda_r \in \{0, 1\} \quad r \in R_v \tag{36}$$

The objective minimizes the costs of purchased bunker, startup costs and slack costs. The parameters  $\underline{w}$  and  $\bar{w}$  denote a penalty for violating the minimal  $\underline{q}_c$  and maximal  $\bar{q}_c$  quantity imposed by contract  $c$ . Constraints (34) ensures that all contracts are fulfilled. Convexity constraints (35) ensure that exactly one bunker pattern is chosen for each vessel.

Due to the large number of columns in the model [32] proposed to solve the LP relaxed model by Column Generation. Using the generated columns from the LP-solution, the resulting problem is solved to integer optimality using a MIP solver, leading to a heuristic solution for the original problem.

Initially all dual variables are set to zero, a subproblem is constructed for each vessel and solved as a MIP problem. The first master problem is then constructed with one solution for each vessel as columns. This master is solved and the first values are found. The subproblems are resolved for all vessels (only the objective coefficients for the contracts needs updating) and new columns are generated for the master. This continues until no negative reduced cost columns can be generated, and the LP optimal solution is achieved.

The subproblems do not need to be solved to optimality since any column with negative reduced cost will ensure progress of the algorithm. Therefore the solver is allowed to return solutions to the subproblem having a considerable optimality gaps. As the algorithm progresses, the allowable subproblem gap is reduced.

A simple form of dual stabilization has been used in the implementation by [32] to speed up convergence. The Box-step method imposes a box around the dual variables, which are limited from changing more than  $\pi_{max}$  per iteration. This has been motivated by the dual variables only taking on values  $\{-w, \bar{w}, 0\}$  in the first iteration, these then stabilize at smaller numerical values in subsequent iterations.

The model is able to solve even very large real-life instances involving more than 500 vessels, 40,000 port calls, and 750 contracts. First, column generation is used to solve the linearized model, and then a MIP solver is used to find an integer solution only using the generated columns. This results in a small gap in the optimal solution compared to if all columns were known. However, computational results show that the gap is never more than around 0.5% even for the largest instances. In practice the resulting gap of the algorithm, can be much smaller since the found solutions are benchmarked against a lower bound and not against the optimal solution.

An interesting side product of the model is the dual variables  $\underline{\pi}_c$  and  $\overline{\pi}_c$  for the upper and lower contract constraints (34). These values can be used to evaluate the gain of a given contract, which may be valuable information when (re)negotiating contracts.

Since bunker prices are stochastic of nature, future research should be focused on modeling the price fluctuation. However, the models tend to become quite complex and difficult to solve as observed by [34], while only adding small extra improvements to the results. So a trade-off must be done between model complexity and gain in bunker costs. The work of [37] shows some promising developments in this important direction.

Also, instruments from finance (bunker future or forward contracts, fixed price bunker fuel swaps) could be used to control risk in bunker purchasing, and to increase the margins on oil trade. Bunker purchasing for liner ships constitutes such a big market that it deserves a professional trading approach.

## 8 The Vessel Schedule Recovery Problem

It is estimated that approximately 70–80 % of vessel round trips experience delays in at least one port. The common causes are bad weather, strikes in ports, congestions in passageways and ports, and mechanical failures.

Currently when a disruption occur, the operator at the shipping companies manually decides what action to take. For a single delayed vessel a simple approach could be to speed up. However, the consumption of bunker fuel is close to a cubic function of speed and vessels' speeds are limited between a lower and upper limit. So even though an expensive speed increase strategy is chosen, a vessel can arrive late for connections, propagating delays to other parts of the network. Having more than 10,000 containers on board a large vessel, calculating the overall consequences of re-routing/delaying these containers demands algorithms for big data. Disruption management is well studied within the airline industry (see [4] or [16] for a review) and the network design of airlines resemble liner shipping networks inspiring the few works on disruption management found for liner shipping. Mulder et al. [27] presents a markov decision model to determine the optimal recovery policy. The core idea is to reallocate buffer time within a schedule in order to recover from disruptions. Brouer et al. [13] present the Vessel Schedule Recovery Problem (VSRP) handling a disruption in a liner shipping network by omitting port calls, swapping port calls or speeding up vessels in a predefined disruption scenario. The model and method will be presented in the following section.

### 8.1 Definitions

A given disruption scenario can mathematically be described by a set of vessels  $V$ , a set of ports  $P$ , and a time horizon consisting of discrete time slots  $t \in T$ . The time slots are discretized on port basis as terminal crews handling the cargo operate in shifts, which are paid for in full, even if arriving in the middle of a shift. Hence we only allow vessels arriving at the beginning of shifts. Reducing the graph to timeslots based on these shifts, also has the advantage of reducing the graph size, although this is a minor simplification of the problem. For each vessel  $v \in V$ , the current location and a planned schedule consisting of an ordered set of port calls  $H_v \subseteq P$  are known within the recovery horizon, a port call  $A$  can precede a port call  $B$ ,  $A < B$  in  $H_v$ . A set of possible sailings, i.e. directed edges,  $L_h$  are said to *cover* a port call  $h \in H_v$ . Each  $L_h$  represent a sailing with a different speed.

The recovery horizon,  $T$ , is an input to the model given by the user, based on the disruption in question. Inter continental services will often recover by speeding during ocean crossing, making the arrival at first port after an ocean crossing a good horizon, severe disruptions might require two ocean crossings. Feeders recovering at arrival to their hub port call would save many missed transshipments giving an

obvious horizon. In combination with a limited geographical dimension this ensures that the disruption does not spread to the entire network.

The disruption scenario includes a set of container groups  $C$  with planned transportation scenarios on the schedules of  $V$ . A feasible solution to an instance of the VSRP is to find a sailing for each  $v \in V$  starting at the current position of  $v$  and ending on the planned schedule no later than the time of the recovery horizon. The solution must respect the minimum and maximum speed of the vessel and the constraints defined regarding ports allowed for omission or port call swaps. The optimal solution is the feasible solution of minimum cost, when considering the cost of sailing in terms of bunker and port fees along with a strategic penalty on container groups not delivered “on-time” or misconnecting altogether.

### 8.2 Mathematical Model

Brouer et al. [13] use a time space graph as the underlying network, but reformulate the model to address the set of available recovery techniques, which are applicable to the VSRP.

The binary variables  $x_e$  for each edge  $e \in E_s$  are set to 1 iff the edge is sailed in the solution. Binary variables  $z_h$  for each port call  $h \in H_v, v \in V$  are set to 1 iff call  $h$  is omitted. For each container group  $c$  we define binary variables  $o_c \in \{0, 1\}$  to indicate whether the container group is delayed or not and  $y_c$  to account for container groups misconnecting. The parameter  $O_e^c \in \{0, 1\}$  is 1 iff container group  $c \in C$  is delayed when arriving by edge  $e \in L_{T_c}$ .  $B_c \in H_v$  is defined as the origin port for a container group  $c \in C$  and the port call where vessel  $v$  picks up the container group. Similarly, we define  $T_c \in H_w$  as the destination port for container group  $c \in C$  and the port call where vessel  $w$  delivers the container group. Intermediate planned transshipment points for each container group  $c \in C$  are defined by the ordered set  $I_c = (I_c^1, \dots, I_c^m)$ . Here  $I_c^i = (h_v^i, h_w^i) \in (H_v, H_w)$  is a pair of calls for different vessels ( $v, w \in V | v \neq w$ ) constituting a transshipment. Each container group  $c$  has  $m^c$  transshipments.  $M_c^e$  is the set of all non-connecting edges of  $e \in L_h$  that result in miss-connection of container group  $c \in C$ .  $M_c \in \mathbb{Z}_+$  is an upper bound on the number of transshipments for container group  $c \in C$ .

Let the demand of vessels  $v$  in a node  $n$  be given by  $S_v^n = -1$  if  $n = n_s^v, S_v^n = 1$  if  $n = n_t^v$ , while  $S_v^n = 0$  for all other nodes. Then we get the following model:

$$\min \sum_{v \in V} \sum_{h \in H_v} \sum_{e \in L_h} c_e^v x_e + \sum_{c \in C} (c_c^m y_c + c_c^d o_c) \tag{37}$$

$$\text{s.t.} \quad \sum_{e \in L_h} x_e + z_h = 1 \quad v \in V, h \in H_v \tag{38}$$

$$\sum_{e \in n^-} x_e - \sum_{e \in n^+} x_e = S_v^n \quad v \in V, n \in N_v \tag{39}$$

$$y_c \leq o_c \quad c \in C \quad (40)$$

$$\sum_{e \in L_{T_c}} O_e^c x_e \leq o_c \quad c \in C \quad (41)$$

$$z_h \leq y_c \quad c \in C, h \in B_c \cup I_c \cup T_c \quad (42)$$

$$x_e + \sum_{\lambda \in M_c^e} x_\lambda \leq 1 + y_c \quad c \in C, e \in \{L_h | h \in B_c \cup I_c \cup T_c\} \quad (43)$$

$$x_e \in \{0, 1\}, e \in E_s \quad y_c, o_c \in \mathbb{R}_+, c \in C \quad z_h \in \mathbb{R}_+, v \in V, h \in H_v \quad (44)$$

The objective function (37) minimizes the cost of operating vessels at the given speeds, the port calls performed along with the penalties incurred from delaying or misconnecting cargo.

Constraints (38) are set-partitioning constraints ensuring that each scheduled port call for each vessel is either called by some sailing or omitted. The next constraints (39) are flow-conservation constraints. Combined with the binary domain of variables  $x_e$  and  $z_h$  they define feasible vessel flows through the time-space network. A misconnection is by definition also a delay of a container group and hence the misconnection penalty is added to the delay penalty, as formulated in (40). Constraints (41) ensure that  $o_c$  takes the value 1 iff container group  $c$  is delayed when arriving via the sailing represented by edge  $e \in E_s$ . Constraints (42) ensure that if a port call is omitted, which had a planned (un)load of container group  $c \in C$ , the container group is misconnected. Constraints (43) are coherence constraints ensuring the detection of container groups' miss-connections due to late arrivals in transshipment ports. On the left-hand side the decision variable corresponding to a given sailing,  $x_e$ , is added to the sum of all decision variables corresponding to having onward sailing resulting in miss-connections,  $\lambda \in M_c^e$ .

In [13] the model has been tested on a number of real-life cases, including a delayed vessel, a port closure, a berth prioritization, and expected congestion. An analysis of the four real life cases, show that a disruption allowing to omit a port call or swap port calls may ensure timely delivery of cargo without having to increase speed and hence, a decision support tool based on the VSRP may aid in decreasing the number of delays in a liner shipping network, while maintaining a slow steaming policy. To operationalize this the rerouting of the actual flow and adjustment of the actual schedule must be incorporated in a real time system to enable here-and-now decisions. This is especially challenging for larger disruption scenarios than the ones described as the size of the problem grows exponentially.

## 9 Conclusion and Future Challenges

Maritime logistics companies operate in an environment which requires them to become more and more analytical. In general there are several insights to be gained from the data companies has available. Especially when companies start to use the forward looking analytical techniques rather than only using data for backward

looking analysis (descriptive and diagnostic models) companies can unlock significant value from the collected data as shown in this chapter. Forward looking techniques (predictive models) can provide input for the decision making process where the best possible action is sought (prescriptive models). A pressing challenge in big data analysis today lies in the integration of predictive and prescriptive methods which combined can serve as valuable decision support tools. This chapter introduced a selection of large scale planning problems within maritime logistics with a primary focus on challenges found in the liner shipping industry. Focus has been on addressing strategic, tactical and operational problems by modern large scale optimization methods. However optimization within maritime logistics is complicated by the uncertainty and difficult accessibility of data. Most demands are only estimates, and for historic reasons even contracted cargo can be unreliable since there are no penalties associated with no-show cargo. To limit these uncertainties predictive machine learning techniques is an important tool. In particular, seasonal variations and similar trends can be predicted quite well and decision support systems should take such uncertainties into account. This can be done either by developing models where it is possible to re-optimize the problem quickly in order to meet new goals and use them interactively for decision support and for evaluating what-if scenarios suggested by a planner as there are still many decisions that will not be data-driven. Quantitative data can not always predict the future well in situations of e.g. one-time events and generally extrapolation is hard. But in situations where we operate in an environment where data can be interpolated mathematical models may serve as great decision support tools by integrating the predictive models directly in the prescriptive model. With the large volume of data generated by carriers, increased quality of forecasts, and algorithmic improvements it may also be beneficial and even tractable to include the uncertainties directly in the decision models. A relatively new way of handling data uncertainty is by introducing uncertainty sets in the definition of the data used for solving large-scale LP's. The standard LP found as a subproblem in many of the described problems can generically be stated as  $\min_x \{c^T x : Ax \leq b\}$ , where  $A$ ,  $b$ , and  $c$  contain the data of the problem at hand. As described previously in this chapter most of the data is associated with uncertainties but in *Robust Optimization* this can be handled by replacing the original LP with an uncertain LP  $\{\min_x \{c^T x : Ax \leq b\} : (A, b, c) \in \mathcal{U}\}$ . The best robust solution to the problem can be found by solving the Robust Counterpart of the problem, which is an semi-infinite LP  $\min_{x,t} \{t : c^T x \leq t, Ax \leq b \forall (A, b, c) \in \mathcal{U}\}$ . Clearly this LP is larger than the original LP, but with good estimates of the uncertainty sets the size can be manageable, further details can be found in [6]. As the accuracy of predictive models increase it will be possible to come up with good estimates for the uncertainty sets and thereby actually making it feasible to solve robust versions of the planning problems. In the MIP case the problems usually become much harder and often intractable with a few exceptions. An alternative approach to Robust Optimization is to handle the uncertainties via probability distributions on the data and use *Stochastic Programming* and solve the chance constrained program  $\min_{x,t} \{t : \text{Prob}_{(A,b,c) \sim P} \{c^T x \leq t, Ax \leq b\} \geq 1 - \epsilon\}$  or a two-stage stochastic program based on a set of scenarios. Again, machine learning algorithms can provide good

estimates of the actual underlying distributions or expected scenarios and it may be possible to obtain results that are less conservative than the worst-case results provided by Robust Optimization, but the process can be more computationally extensive.

## References

1. Agarwal, R., Ergun, Ö.: Ship scheduling and network design for cargo routing in liner shipping. *Transp. Sci.* **42**, 175–196 (2008)
2. Alvarez, J.F.: Joint routing and deployment of a fleet of container vessels. *Marit. Econ. Logist.* **11**, 186–208 (2009)
3. Archetti, C., Speranza, M.G.: A survey on matheuristics for routing problems. *EURO J. Comput. Optim.* **2**, 223–246 (2014)
4. Ball, M., Barnhart, C., Nemhauser, G., Odoni, A.: Chapter 1: Air Transportation—Irregular Operations and Control. In: *Handbooks in Operations Research and Management Science: Transportation*, vol. 14. Elsevier (2007)
5. Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W., Vance, P.H.: Branch-and-price: column generation for solving huge integer programs. *Oper. Res.* **46**(3), 316–329 (1998)
6. Ben-Tal, A., Nemirovski, A.: Robust optimization-methodology and applications. *Math. Program.* **92**(3), 453–480 (2002)
7. Bertsimas, D., King, A., Mazumder, R.: Best subset selection via a modern optimization lens. Submitted to *Annals of Statistics* (2014)
8. Bertsimas, D., Tsitsiklis, J.N.: *Introduction to Linear Optimization*, vol. 6. Athena Scientific, Belmont (1997)
9. Besbes, O., Savin, S.: Going bunkers: the joint route selection and refueling problem. *Manuf. Serv. Oper. Manag.* **11**, 694–711 (2009)
10. 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**, 281–312 (2014a)
11. Berit D. Brouer, J. Fernando Alvarez, Christian E. M. Plum, David Pisinger, Mikkel M. Sigurd 'A Base Integer Programming Model and Benchmark Suite for Liner-Shipping Network Design' *Transportation Science*, **48**(2), pp. 281–312 (2014)
12. Brouer, B., Desaulniers, G., Pisinger, D.: A matheuristic for the liner shipping network design problem. *Transp. Res. Part E Logist. Transp. Rev.* **72**, 42–59 (2014b)
13. Brouer, B., Dirksen, J., Pisinger, D., Plum, C., Vaaben, B.: The vessel schedule recovery problem (VSRP) a MIP model for handling disruptions in liner shipping. *Eur. J. Oper. Res.* **224**, 362–374 (2013)
14. Brouer, B., Pisinger, D., Spoorendonk, S.: Liner shipping cargo allocation with repositioning of empty containers. *INFOR: Inf. Syst. Oper. Res.* **49**, 109–124 (2011)
15. Burke, E.K., Kendall, G.: *Search Methodologies*. Springer (2005)
16. Clausen, J., Larsen, A., Larsen, J., Rezanova, N.J.: Disruption management in the airline industry-concepts, models and methods. *Comput. Oper. Res.* **37**(5), 809–821 (2010)
17. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., et al.: *Introduction to Algorithms*, vol. 2. MIT Press, Cambridge (2001)
18. Costa, A.M.: A survey on benders decomposition applied to fixed-charge network design problems. *Comput. Oper. Res.* **32**(6), 1429–1450 (2005)
19. Delgado, A., Jensen, R.M., Janstrup, K., Rose, T.H., Andersen, K.H.: A constraint programming model for fast optimal stowage of container vessel bays. *Eur. J. Oper. Res.* **220**(1), 251–261 (2012)
20. Desrosiers, J., Lübbecke, M.E.: *A Primer in Column Generation*. Springer (2005)
21. Erera, A.L., Morales, J., Savelsbergh, M.: Global intermodal tank container management for the chemical industry. *Transp. Res. Part E* **41**(6), 551–566 (2005)



22. Gendreau, M., Potvin, J.-Y.: *Handbook of Metaheuristics*, vol. 2. Springer (2010)
23. Gendron, B., Crainic, T.G., Frangioni, A.: *Multicommodity Capacitated Network Design*. Springer (1999)
24. Karsten, C.V., Pisinger, D., Ropke, S., Brouer, B.D.: The time constrained multi-commodity network flow problem and its application to liner shipping network design. *Transp. Res. Part E Logist. Transp. Rev.* **76**, 122–138 (2015)
25. Kim, H.-J., Chang, Y.-T., Kim, K.-T., Kim, H.-J.: An epsilon-optimal algorithm considering greenhouse gas emissions for the management of a ships bunker fuel. *Transp. Res. Part D Transp. Environ.* **17**, 97–103 (2012)
26. Meng, Q., Wang, S.: Liner shipping service network design with empty container repositioning. *Transp. Res. Part E Logist. Transp. Rev.* **47**(5), 695–708 (2011)
27. Mulder, J., Dekker, R., Sharifyazdi, M.: *Designing robust liner shipping schedules: optimizing recovery actions and buffer times*. Report/Econometric Institute, Erasmus University Rotterdam (2012)
28. Pacino, D.: *Fast generation of container vessel stowage plans*. Ph.D. thesis, IT University of Copenhagen (2012)
29. Pacino, D., Delgado, A., Jensen, R., Bebbington, T.: Fast generation of near-optimal plans for eco-efficient stowage of large container vessels. *Comput. Logist.* **6971**, 286–301 (2011)
30. Pacino, D., Jensen, R.: Constraint-based local search for container stowage slot planning. *Lect. Notes Eng. Comput. Sci.* **2**, 1467–1472 (2012)
31. Pacino, D., Jensen, R.: Fast slot planning using constraint-based local search. *IAENG Trans. Eng. Technol.* **186**, 49–63 (2013)
32. Plum, C., Pisinger, D., Jensen, P.: Bunker purchasing in liner shipping. *Handbook of Ocean Container Transport Logistics*, vol. XVII, pp. 251–278. Springer (2015)
33. Plum, C., Pisinger, D., Sigurd, M.M.: A service flow model for the liner shipping network design problem. *Eur. J. Oper. Res.* **235**(2), 378–386 (2014)
34. Plum, C.E.M., Jensen, P.N.: *Minimization of bunker costs*. Master thesis, University of Copenhagen (2007)
35. Reinhardt, L.B., Pisinger, D.: A branch and cut algorithm for the container shipping network design problem. *Flex. Serv. Manuf. J.* **24**(3), 349–374 (2012)
36. Rossi, F., Van Beek, P., Walsh, T.: *Handbook of Constraint Programming*. Elsevier (2006)
37. Sheng, X., Lee, L., Chew, E.: Dynamic determination of vessel speed and selection of bunkering ports for liner shipping under stochastic environment. *OR Spectr.* **36**(2), 455–480 (2014)
38. Shintani, K., Imai, A., Nishimura, E., Papadimitriou, S.: The container shipping network design problem with empty container repositioning. *Transp. Res. Part E Logist. Transp. Rev.* **43**(1), 39–59 (2007)
39. Song, D.-P., Dong, J.X.: Cargo routing and empty container repositioning in multiple shipping service routes. *Transp. Res. Part B Methodol.* **46**(10), 1556–1575 (2012)
40. Tierney, K., Askelsdottir, B., Jensen, R.M., Pisinger, D.: Solving the liner shipping fleet repositioning problem with cargo flows. *Transp. Sci.* **49**(3), 652–674 (2015)
41. Toth, P., Vigo, D.: *Vehicle Routing: Problems, Methods, and Applications*. SIAM (2014)
42. Vielma, J.P.: Mixed integer linear programming formulation techniques. *SIAM Rev.* **57**(1), 3–57 (2015)
43. Wang, S., Meng, Q.: Robust bunker management for liner shipping networks. *Eur. J. Oper. Res.* **243**(3), 789–797 (2015)
44. Yao, Z., Ng, S.H., Lee, L.H.: A study on bunker fuel management for the shipping liner services. *Comput. Oper. Res.* **39**, 1160–1172 (2012)

## Author Biographies



**Berit Dangaard Brouer** holds a Ph.D. in operations research from the Technical University of Denmark. Her research targets maritime logistics within the liner shipping industry. She has several publications in internationally recognized journals and her publications propose decision support tools for a range of planning problems within network design, cargo routing and disruption management.



**Christian Vad Karsten** is a Ph.D. student at the Technical University of Denmark where he is working on computational methods for large-scale optimization problems. His research is focused on advanced analytical methods and the application of these to data-driven problems to help make better decisions. More specifically he is working on how to introduce competitiveness in mathematical models addressing big data planning problems faced by container shipping companies. The project is part of a close cooperation between academia and industry aimed at developing effective models and methods to support planning and managerial decision-making. In addition to his mathematical training at the Technical University of Denmark, Christian has been a visiting student at Princeton University, UC Berkeley, and Massachusetts Institute of Technology.



**David Pisinger** is professor and head of research in Management Science, Technical University of Denmark (DTU). His principal focus of research is Maritime Optimization, Railway Optimization, and recently, Energy Models. In his research he tries to combine theoretical results with real-life applications. He has been principal investigator of the research projects ENERPLAN and GREENSHIP on maritime optimization in collaboration with Maersk Line. Professor Pisinger received the Ph.D. prize of the Danish Academy of Natural Sciences in 1995, and the award of teaching excellence from Faculty of Natural Sciences, University of Copenhagen in 2000. In 2013 he received Hedorf Fond prize for Transportation Research. He has been nominated as “Teacher of the year” at DTU several times, and several of his students have been awarded prizes for their master and Ph.D. theses.