



A matheuristic for the Cargo Mix Problem with Block Stowage

Christensen, Jonas Mark; Pacino, Dario

Published in:

Transportation Research. Part E: Logistics and Transportation Review

Link to article, DOI:

[10.1016/j.tre.2016.10.005](https://doi.org/10.1016/j.tre.2016.10.005)

Publication date:

2017

Document Version

Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):

Christensen, J. M., & Pacino, D. (2017). A matheuristic for the Cargo Mix Problem with Block Stowage. *Transportation Research. Part E: Logistics and Transportation Review*, 97, 151-171. <https://doi.org/10.1016/j.tre.2016.10.005>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

A Matheuristic for the Cargo Mix Problem with Block Stowage

Jonas Christensen^{*1} and Dario Pacino^{†1}

¹*Technical University of Denmark, Department of Management*

Abstract

The cargo-mix problem aims at selecting the amount of containers of a given type to load on a vessel. In this article we present an extended definition that includes the analysis of a circular route with draft restrictions, limitations on expected cargo and the use of a block stowage strategy. A compact formulation of the problem based on the state-of-the-art heuristic decomposition is shown not to be able to solve the extended problem, thus a matheuristic approach is presented that can achieve high quality results in a matter of seconds.

1 Introduction

Aside from the few years of financial crisis, the liner shipping industry has had a continuous growth. The demand for efficient and cheap transportation and a fierce competition has driven shipping rates down, making it paramount for the carriers to utilise their vessels as efficiently as possible.

Focus on vessel intake maximisation is no news for the shipping lines. Container vessels are delivered with a nominal capacity that ship owners know is only theoretical. Unless the cargo weight distribution is perfect, the nominal intake cannot be reached. Stowage coordinators fight this battle every day. They are the planners of the cargo and have to find a load configuration (*stowage plan*) that both suits the current cargo to be loaded but also guarantees that the vessel can be utilised to its maximum in future ports. The size of nowadays vessels is, however, making this work harder (Pacino et al., 2011). Moreover, the cargo composition available in the different ports might not be suitable for the full utilisation of the vessel. To give a brief example, consider a vessel that has to load a high number of very heavy containers. As a consequence, the *draft* of the vessel (the distance between the waterline and the bottom of the hull) will be greater. If the vessel is scheduled to visit a port with a low water depth, stowage coordinators will have to leave a number of containers behind to reduce the draft and avoid the risk of running the vessel aground. Leaving containers behind is clearly not a desirable situation.

The focus of our work is the analysis of vessels' cargo mix (*the cargo mix problem*), in particular finding what cargo composition is needed for a vessel to maximise its utilisation on a given service. Differing from stowage planning, where a list of pre-selected containers must be stowed on the vessel, the cargo mix problem aims at selecting the quantity of containers of each type that should be loaded on a vessel to maximise its intake. Moreover, standard stowage planning approaches only consider the current port, whereas the cargo mix problem considers a cyclic service with multiple ports. The proposed model can have multiple applications ranging from driving rates prices, improving fleet composition and network design (Christiansen et al., 2007; Reinhardt and Pisinger, 2012; Brouer et al., 2014), or analyse the difference between an expected cargo load and an optimal one.

^{*}Electronic address: jomc@dtu.dk ; Corresponding author

[†]Electronic address: darpa@dtu.dk

In the work of Delgado (2013), it was shown that a cargo-mix analysis based on simple capacity limitations greatly overestimates the revenue of the vessel. The presented model, however, did not include a number of features which we deem essential for a correct capacity or revenue calculation.

In our work, we propose an extension of the cargo-mix problem that includes draft restrictions at the visited ports and a block stowage strategy. This last refers to the logical partitioning of the vessel into blocks. Each block is then allowed to host only containers that have the same discharge port. This alternative way of stowing containers is aimed at improving operations at ports, since it makes it possible to perform e.g. dual cycling (where load and discharge operations are no longer sequential). Examples of block stowage are found in the stowage planning literature in e.g. Ambrosino et al. (2015b).

Our work contributes to the state-of-the-art in the following way. First we extend the formal definition of the cargo mix problem from Delgado (2013) to include block stowage, circular routes, draft restrictions, and cargo flow limitations. Second, we propose an extension of the compact formulation of Delgado (2013) where the new problem features are included. We also show that the new model can no longer be solved via mixed-integer programming. At last, we propose and compare a number of matheuristics that are able to solve the problem in a matter of seconds.

The remainder of the paper is organised as follows. Section 2 presents the necessary background knowledge. In Section 3, the existing relevant literature is reviewed. Section 4 describes the problem in deeper details and presents a compact formulation for the problem. Section 5 describes the matheuristic approach. Section 6 is a brief explanation of the data used, and Section 7 presents the results for the MIP model as well as the matheuristic. Lastly Section 8 contains final remarks and conclusions.

2 Background

Optimising the intake of a vessel requires deep understanding of vessel architecture, and the industry as a whole. Liner shipping companies transport containers between ports on a fixed cyclic schedule. Most containers transported on liner vessels are 8 feet wide, 8'6" high and 20', 40' or 45' long. *High-cube* containers also exist, which are 9'6" high. Some containers have an integrated cooling unit for the transport of e.g. perishable goods. These containers are called *reefers* and must be plugged into an electrical outlet, and thus can only be stowed where such a plug is available. Additionally, there are dangerous goods containers (*IMO containers*) which must follow strict segregation rules.

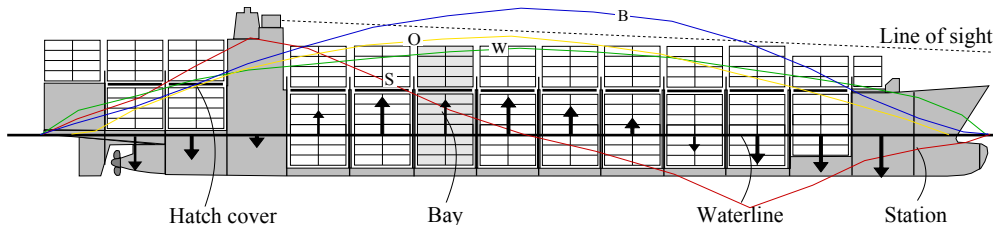


Figure 1: Example figure of a container vessel.

Figure 1 shows the layout of a container vessel. A vessel is divided into *bays*. Each bay is divided into an *on-deck* and *below-deck* part, physically separated by a *hatch cover*. Hatch covers are flat leak proof structures that can be removed when loading/unloading containers. Each bay consists of a number of *stacks* divided into *slots*. A slot can hold a single 20' container. Figure 2a shows a transversal section of a bay, where the numbers denote the blocks, and Figure 2b shows

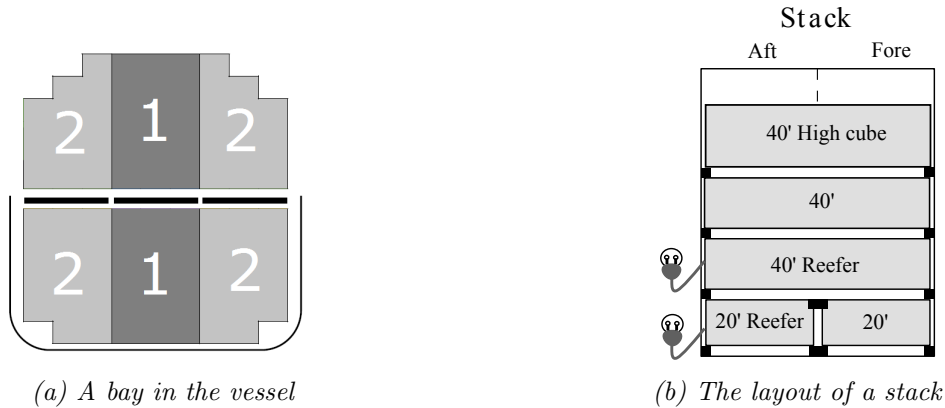


Figure 2: Example figure showing the design of a bay, and a stack.

the design of a stack within a bay. Two slots side by side are denoted a *cell*, each consisting of an *aft* (towards the stern) and a *fore* (towards the bow) slot. Each stack has weight and height limits. A stack has outer and inner supports and, as shown in Figure 2b, a 20' container uses the inner and the outer supports, whereas a 40' container is supported only by the outer supports. A stack thus has two weight limits, one for the outer supports and one for the inner supports. Due to the construction of the vessel the inner supports are weaker than the outer supports. Twist locks can be used to lock containers to each other at the corners. There are no such locking mechanisms in the middle of 40' containers, hence stacking rules impose that 20' containers cannot be stowed above 40' containers. Twist locks have limited strength, thus container weights must normally decrease upwards in stacks on deck. Below deck, cell guides secure containers in place.

Vessels capacity is measured in *Twenty-foot Equivalent Units* (TEU), a 40' standard and 40' high-cube container are 2 TEU, or 1 *FEU* (*Forty-foot Equivalent Unit*). In liner shipping profit comes from the transported containers, and naturally, carriers want to utilise the full capacity to increase revenue. The main expenses are fuel cost and costs associated with port stays. Reducing the duration of a port stay can be done by e.g. reducing *overstowage*. Overstowage happens when a container is stowed below a container destined for a later port. Containers can overstay in two different ways. *Stack overstowage* occurs when the overstowing container is in the same stack as the overstowed container and *hatch overstowage* occurs when a container on deck is overstowing a container below deck. If hatch overstowage occurs all containers above the hatch cover must be restowed, in order to unload the overstowed container.

A current trend that is becoming popular in the industry, is the notion of *block stowage*. To facilitate better planning and ease port handling operations, the ship is divided into blocks (a logical grouping of stacks). Containers in a block must, as a rule of thumb, have the same discharge port. There can be different definitions of a block, often company dependent. Here, according to our industrial collaborator, stacks below and above the same hatch-cover belong to the same block. This means that, given that all containers within the block must have the same discharge port, overstowage and hatch-overstowage cannot happen. Moreover, as it can also be seen in Figure 2a, external blocks are seen as one to ease transverse stability (as modelled in e.g. Pacino et al. (2011)).

Any stowage plan must result in a seaworthy vessel. A vessel is declared seaworthy if its static stability is correct and all stress forces are within limits. Stress limits are only known at specific points across the length of the vessel, these calculation points are called *frames*. Figure 1 shows a longitudinal section of a vessel with 14 frames. The *W* curve shows the weight distribution across the length of the vessel, and the *O* curve represents the buoyancy force. The arrows show the resulting force acting on each of the sections. If these sections were allowed to move freely as independent objects, some of them would sink deeper into the water and some would rise depending on the resulting force acting on the section. However, they are not allowed to move freely, and

thus this causes shearing and bending stresses over the vessel structure. The shearing and bending limits are known for every frame. These limits must be satisfied when leaving each port. Shear force is shown in Figure 1 as the curve S , and the bending moment is the curve B .

Most container vessels are equipped with *ballast tanks*. These tanks can be used to load water in order to achieve a better weight distribution, reduce stress forces, or fix stability issues. However, the use of these tanks is discouraged as this would increase the total weight (*displacement*) of the vessel, thus increasing fuel consumption.

Additional stability limits must be held for the ship to be declared seaworthy, e.g. *trim*. The *aft draft* is the distance between the waterline and the bottom of the hull at the stern of the ship, and the *fore draft* is at the bow. Trim is defined as the difference between the aft draft and the fore draft, and must be within its limits. Moreover, the height a stack can reach is limited by the *line of sight*. The captain standing on the bridge must be able to see a distance equal to the double of the length of the vessel, or 500 meters, whichever is smallest. The line of sight depends on the draft of the vessel and is thus not simply a stack capacity reduction. The *metacentric height* describes the vessel's ability to return to its initial position when affected by an external force. Figure 3 shows the transversal section of a vessel. The metacentric height is defined as the distance between the point M (*metacentre*) and the point G (the centre of gravity). The metacentric height must be within operational limits to ensure that the vessel will not capsize.

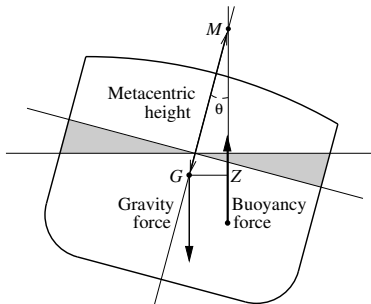


Figure 3: Metacentric height of a vessel

If the described stacking rules, stability, and stress limits are held at bay then the vessel can be declared seaworthy, and can hereafter safely depart.

A vessel is fully described by a document called *vessel profile*, which specifies all the weight and capacity limits, a general outline of the vessel etc. The non-linear functions for the draft, centre of gravity, trim and metacentre are outlined in the *hydrostatic table*, within the vessel profile. We refer the reader to Delgado (2013) for a more detailed description of the stability requirements and stacking rules.

3 Literature Review

Academic work on the liner shipping cargo mix problem is limited. The first formal description was presented in the PhD thesis of Delgado (2013) where an integer programming model was presented. The author shows that for multi-port scenarios the model does not scale. To achieve scalability, the same decomposition as in earlier stowage planning work (Pacino et al., 2011) was applied. The results show that the decomposition gives an accurate estimate. Tested on 360 instances, on a service of up to 10 ports, the approach was able to optimally solve 91.7% of the instances, with a time limit of 60 minutes. In comparison to ours, the work in Delgado (2013) is only able to analyse a portion of a service and not a full rotation. Moreover, there is no restriction to the number of containers that can be selected, whereas our version of the problem is bounded to the expected

cargo-flows of the analysed service, and imposes draft restrictions on the visited ports. Our work also shows that the inclusion of a block stowage strategy makes the MIP model proposed in Delgado (2013) intractable for industrial size instances.

In Zurheide and Fischer (2011) the authors presents a slot allocation model which helps the carrier decide whether to accept or reject a booking request. The decision takes into account the possibility to postpone a container in order to accept a more beneficial one. A revenue model for a short-sea shipping service is also presented in Feng and Chang (2008). The model disregards major aspects of stowage planning and only consider TEU and weight capacity. In Delgado (2013), it was shown that this model due to the inaccuracies, in average, overestimates the revenue by 8%. In this work, we fill the gaps in the literature by describing a fast and scalable heuristic that considers cargo flows, circular routes and draft at port. Inspired by previous work we solve a master problem based on stowage planning as Delgado (2013) has shown this can be used to accurately approximate the revenue. In the future, we envision to extend this work by considering stochastic cargo flows, and the extension of the analysis to multiple vessels.

Given the limited amount of academic work on liner shipping cargo-mix problems, we find it relevant to introduce the literature related to stowage planning since the Cargo Mix Problem with Block Stowage (CMPBS) could be seen as a generalisation. The main difference between the CMPBS and stowage planning is that CMPBS performs cargo selection based on a set of expected cargo flows between the visited ports, whereas in stowage planning a list of containers to load has already been selected. Stowage planning is mainly concerned with the loading of the containers at the current port; in contrast, CMPBS analyses a cyclic route. In the work of Pacino et al. (2012) it was shown that vessel stability constraints are non-linear if the cargo weight is variable, which is the case in the CMPBS. These non-linearities, however, can be approximated with a small number of linear planes without losing too much information, as described in Delgado (2013). Botter and Brinati (1992) presents an accurate formulation which includes metacentric height, transversal stability, trim, shear forces, and bending moment as constraints. The model fails to solve for small vessels with 1000 TEU capacity, and therefore a heuristic decomposition is proposed. Integer programming models assigning containers to slots are also presented in Ambrosino et al. (2004) and Li et al. (2008). The main stability consideration enforced in these models is that the left and right weight difference and bow and stern weight difference must be within a certain predefined limit. However, scalability issues of these models make them unsuitable for practical use. Integer programming models for stowage planning, assigning containers to specific slots, experience scalability issues due to the large number of variables and constraints. Scalable approaches are based on multi-phase methods. Both Kang and Kim (2002) and Wilson and Roach (2000) describes a 2-phased approach, where the former is of an iterative nature, and in the latter the flow of information only goes downward. A three-step heuristic is introduced in Ambrosino et al. (2009), in which the left and right, and bow and stern weight differences are kept within a tolerance. Pacino et al. (2011) presents a solution method which takes most of the stability issues into consideration. The approach is a 2-phased method where first containers are distributed to subsections of the vessel ensuring that the metacentric height, draft, trim and shear forces are within limits. In the second phase containers are then assigned to specific position. The most accurate published model is, however, the one described in Pacino et al. (2012) which, along with the stability and stress limits considered in Pacino et al. (2011), also includes direct linearizations of hydrostatic data and the modelling of ballast tanks.

Ambrosino et al. (2010) compares an ant colony heuristic and a tabu search heuristic, and Avriel et al. (1998) presents a model that minimise the number of shifts (similar to overstowage) in stowage planning. However, the model has limited applicability due to scalability issues, and therefore the authors developed the Suspensory Heuristic. The Suspensory Heuristic only focuses on container stacking constraints and does not include any stability considerations. Dubrovsky et al. (2002) presents a genetic algorithm (GA) for the minimum shifts problem. The GA obtain results similar

to those obtained by the Suspensory Heuristic but has the possibility of incorporating additional constraints, such as simple stability constraints. Ding and Chou (2015) considers the same problem and describes a state-of-the-art heuristic that improves the results from the Suspensory Heuristic. Sciomachen and Tanfani (2003) uses a 3D-bin packing heuristic to tackle the master bay planning problem. Constraint programming approaches worth mentioning for stowage planning includes Ambrosino and Sciomachen (1998), Delgado et al. (2012) and the constraint-based local search method in Pacino and Jensen (2012).

Of interest is also the work of Ambrosino et al. (2015a,b) that tackles the Multi Port Master Bay Planning Problem (MP-MBPP), where the authors include block stowage constraints similar to those of the CMPBS. The vessel is divided into sections according to their hatch covers, and containers in the same section must have the same destination. In the MP-MBPP all containers must be loaded, and the objective minimises a weighted sum of the crane makespan and number of hatch-overstowing containers. In Ambrosino et al. (2015a) the problem is thoroughly described, and formulated as a mixed integer program. In Ambrosino et al. (2015b) a heuristic based on fixing variables in the MIP model is presented, and it is shown to outperform the MIP model for the 16 tested instances. The method is tested on ships of varying size and using a service of 6 ports. The MP-MBPP considers a sequence of ports, and not a cyclic route as in the CMPBS. Furthermore, the MP-MBPP uses a fixed load list whereas an expected cargo flow is used as input to the CMPBS. Also, the CMPBS imposes a higher degree of accuracy wrt. the stability of the vessel. These differences in the two problems make the heuristic in Ambrosino et al. (2015b) unsuitable for the CMPBS.

4 The Cargo Mix Problem

Given a liner service, a vessel profile and the expected cargo flows, the liner shipping CMPBS aims at finding a mixture of container types to ideally load on the vessel at each port of call. The cargo mix must fulfil the stability requirements and respect the expected cargo flows while optimising a function of cargo values, which in this work is represented by the intake and the revenue. The block stowage requirement is strictly enforced in the CMPBS, thus the containers in each block must have the same discharge port.

The problem is decomposed similarly to earlier work (Kang and Kim, 2002; Wilson and Roach, 2000; Ambrosino et al., 2010; Pacino et al., 2011; Delgado, 2013), but instead of assigning specific containers to a specific slot, containers are divided into *container types*, and container types are assigned to blocks. Each container type represents a number of containers with the same properties in terms of weight, height, length and reefer capabilities. The decomposition is illustrated in Figure 4. In the master planning phase, a number of containers from a container type are assigned to a block while satisfying high-level capacity constraints. The slot planning phase considers each block independently and assigns a specific slot for each of the containers assigned in the master planning phase. In this work, we disregard the slot planning phase since, as Delgado (2013) have shown, the master planning phase gives an accurate estimation.

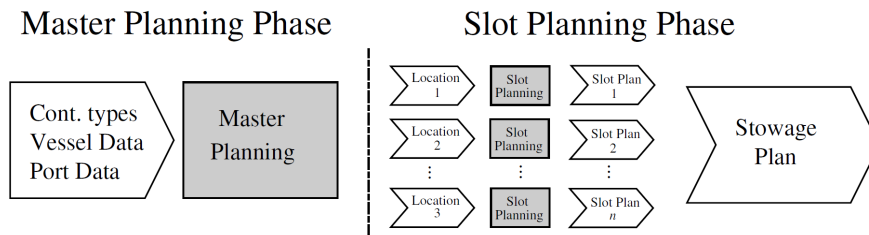


Figure 4: The two-phase heuristic decomposition

The objective of the CMPBS is not to generate fully feasible stowage plans, but to evaluate the capacity and utilisation of a vessel, given a service. We consider the most standard containers: 20' dry and reefer containers, 40' dry and reefer containers both normal height and high-cube. The vessel is enforced to be at even keel (with a trim of zero), and all the previously described stability and stress limits must be held at bay.

The Cargo Mix Problem with Block Stowage can be modelled as a mixed integer program. First, define \mathcal{C} as the set of container types. A container type, $c \in \mathcal{C}$, is defined by the dimensions and properties of the containers and the parameters Γ^c , Φ^c , v^c describes respectively the TEU coefficient (1 for a 20' container, 2 for a 40' container), volume and weight of container type $c \in \mathcal{C}$. The volume is needed for the modelling of the high-cube containers, which need further restrictions than a simple capacity constraint. Also define the set $\mathcal{R} \subset \mathcal{C}$, as the set of all reefer container types, and let the set \mathcal{B} be the set of blocks. In the CMPBS a full cyclic service is considered and \mathcal{P} is the set of ports visited in one rotation. The container demand is described by the set \mathcal{T} and the parameter a^{tc} . The set \mathcal{T} is the set of transports. A transport describes an origin-destination pair and a^{tc} is the number of containers available of type $c \in \mathcal{C}$ during transport $t \in \mathcal{T}$. The set \mathcal{T}_p^{ON} is the set of transports where port p is visited between the origin and destination of the transport. Define P_i^j as the ports visited between i and j , including i but excluding j , then \mathcal{T}_p^{ON} is defined as follows

$$\mathcal{T}_p^{ON} = \{t \in \mathcal{T} \mid p \in P_{o^t}^{d^t}\} \quad \forall p \in \mathcal{P}$$

Where o^t and d^t is the origin and destination port of transport t . Thus for a given port p the set \mathcal{T}_p^{ON} consists of those transports that visit port p on the route from its origin to destination. As an example, Figure 5 shows a graph of a vessel rotation, the possible transports and the transports that are in the sets \mathcal{T}_a^{ON} , \mathcal{T}_b^{ON} , \mathcal{T}_c^{ON} .

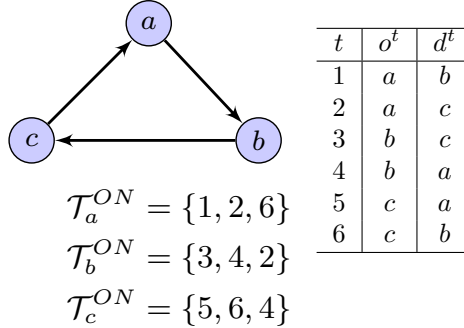


Figure 5: Example illustration of the sets \mathcal{T}_p^{ON}

The main decision variable is $y_b^{tc} \in \mathbb{N}$ defining the number of containers of container type $c \in \mathcal{C}$ of transport $t \in \mathcal{T}$ to be stowed in block $b \in \mathcal{B}$. Furthermore, to ensure that the containers in a block have the same destination port, at each port we need to assign a destination for each block. This decision is captured in the binary variable σ_{bp}^d , defined as follows

$$\sigma_{bp}^d = \begin{cases} 1, & \text{If block } b \in \mathcal{B} \text{ is assigned destination } d \in \mathcal{P} \text{ at port } p \in \mathcal{P} \\ 0, & \text{Otherwise.} \end{cases}$$

Lastly, let $w_{bp} \in \mathbb{R}^+$ denote the total weight stowed in block $b \in \mathcal{B}$ at port $p \in \mathcal{P}$.

The main contribution of this paper is not the modelling of the stability constraints. Hence, to simplify the forthcoming model, we define the polyhedron \mathcal{W} as the weight allocations that results in a stable vessel.

$$\mathcal{W} = \left\{ \mathbf{w} \in \mathbb{R}^{|\mathcal{B}| \cdot |\mathcal{P}|} \text{ s.t. (24) - (43)} \right\}$$

In essence, the polyhedron \mathcal{W} , is a set of constraints ensuring the seaworthiness of the vessel. The constraints are described in Appendix A, and are based on the work by Delgado (2013) which also gives a more thorough description. This simplifies the forthcoming model, and keeps the focus on our contributions, that is, the addition of the block stowage requirement, circular routes and cargo flow limitations.

Below, all the sets, variables, and parameters are summarised, and additional sets and parameters are introduced.

Sets:

\mathcal{P}	Set of ports
\mathcal{T}	Set of transports
\mathcal{T}_p^{ON}	Set of transports that visits port $p \in \mathcal{P}$
\mathcal{C}	Set of container types
$\mathcal{C}^{20} \subset \mathcal{C}$	Set of container types with length 20 feet.
$\mathcal{R} \subset \mathcal{C}$	Set of reefer container types
$\mathcal{R}^{20} \subset \mathcal{R}$	Set of reefer container types, with length 20 feet
\mathcal{B}	Set of blocks.

Variables:

$y_b^{tc} \in \mathbb{N}$	Number of containers of type $c \in \mathcal{C}$ of transport $t \in \mathcal{T}$ to be stowed in block $b \in \mathcal{B}$.
$w_{bp} \in \mathbb{R}^+$	Weight of the containers stowed in block $b \in \mathcal{B}$ at port $p \in \mathcal{P}$.
$\sigma_{bp}^d \in \{0, 1\}$	Variable denoting whether or not block $b \in \mathcal{B}$ can only contain containers with destination $d \in \mathcal{P}$ at port $p \in \mathcal{P}$.

Parameters:

$f^{tc} \in \mathbb{R}^+$	The value of container type $c \in \mathcal{C}$ of transport $t \in \mathcal{T}$.
$u^{tc} \in \mathbb{R}^+$	Revenue of container type $c \in \mathcal{C}$ of transport $t \in \mathcal{T}$.
$p^t \in \mathbb{N}$	Number of ports visited by transport $t \in \mathcal{T}$.
$d^t \in \mathcal{P}$	Destination port for transport $t \in \mathcal{T}$.
$k_b^{TEU} \in \mathbb{N}$	Teu capacity for block $b \in \mathcal{B}$
$k_b^{20} \in \mathbb{N}$	Capacity for 20' containers for block $b \in \mathcal{B}$
$r_b^{Cell} \in \mathbb{N}$	Reefer cell capacity of block $b \in \mathcal{B}$
$r_b^{Slot} \in \mathbb{N}$	Reefer slot capacity of block $b \in \mathcal{B}$
$\Gamma^c \in \{1, 2\}$	TEU coefficient of container type $c \in \mathcal{C}$
$h_b \in \mathbb{R}^+$	Volume capacity for block $b \in \mathcal{B}$
$\Phi^c \in \mathbb{R}^+$	Volume coefficient of container type $c \in \mathcal{C}$
$v^c \in \mathbb{R}^+$	Weight of container type $c \in \mathcal{C}$
$q_b \in \mathbb{R}^+$	Weight capacity of block $b \in \mathcal{B}$
$a^{tc} \in \mathbb{N}$	Number of available containers of type $c \in \mathcal{C}$ for transport $t \in \mathcal{T}$.

With this, the model can be formulated as seen below.

$$\text{Max } Z = \sum_{b \in \mathcal{B}} \sum_{t \in \mathcal{T}} \sum_{c \in \mathcal{C}} f^{tc} y_b^{tc} \quad (1)$$

Subject to:

$$\mathbf{w} \in \mathcal{W} \quad (2)$$

$$\sum_{t \in \mathcal{T}_p^{ON}} \sum_{c \in \mathcal{C}} v^c y_b^{tc} = w_{bp} \quad \forall b \in \mathcal{B}, p \in \mathcal{P} \quad (3)$$

$$\sum_{t \in \mathcal{T}_p^{ON}} \sum_{c \in \mathcal{C}^{20}} y_b^{tc} \leq k_b^{20} \quad \forall b \in \mathcal{B}, p \in \mathcal{P} \quad (4)$$

$$\sum_{t \in \mathcal{T}_p^{ON}} \sum_{c \in \mathcal{C}} \Gamma^c y_b^{tc} \leq k_b^{TEU} \quad \forall b \in \mathcal{B}, p \in \mathcal{P} \quad (5)$$

$$\sum_{t \in \mathcal{T}_p^{ON}} \sum_{c \in \mathcal{R}^{20}} y_b^{tc} \leq r_b^{Slot} \quad \forall b \in \mathcal{B}, p \in \mathcal{P} \quad (6)$$

$$\sum_{t \in \mathcal{T}_p^{ON}} \sum_{c \in \mathcal{R}} \frac{1}{2} \Gamma^c y_b^{tc} \leq r_b^{Cell} \quad \forall b \in \mathcal{B}, p \in \mathcal{P} \quad (7)$$

$$\sum_{t \in \mathcal{T}_p^{ON}} \sum_{c \in \mathcal{C}} \Phi^c y_b^{tc} \leq h_b \quad \forall b \in \mathcal{B}, p \in \mathcal{P} \quad (8)$$

$$\sum_{d \in \mathcal{P}} \sigma_{bp}^d = 1 \quad \forall b \in \mathcal{B}, p \in \mathcal{P} \quad (9)$$

$$y_b^{tc} \leq a^{tc} \sigma_{bp}^d \quad \forall b \in \mathcal{B}, c \in \mathcal{C}, p \in \mathcal{P}, t \in \mathcal{T}_p^{ON} \quad (10)$$

$$\sum_{b \in \mathcal{B}} y_b^{tc} \leq a^{tc} \quad \forall t \in \mathcal{T}, c \in \mathcal{C} \quad (11)$$

$$0 \leq w_{bp} \leq q_b \quad \forall b \in \mathcal{B}, p \in \mathcal{P} \quad (12)$$

$$y_b^{tc} \in \mathbb{N} \quad \forall b \in \mathcal{B}, t \in \mathcal{T}, c \in \mathcal{C} \quad (13)$$

$$\sigma_{bp}^d \in \{0, 1\} \quad \forall b \in \mathcal{B}, d \in \mathcal{P}, p \in \mathcal{P} \quad (14)$$

Objective (1) maximises the value of the containers loaded, where f^{tc} is a general function describing the value of each loaded container. This parameter can be changed to accommodate different objectives. In the case of revenue optimisation, it is

$$f^{tc} = u^{tc} \quad (15)$$

where u^{tc} is the revenue of container type $c \in \mathcal{C}$ of transport $t \in \mathcal{T}$ (see Section 6 for a description on how the revenue data is generated). In the case of intake optimisation, the function is

$$f^{tc} = \Gamma^c p^t \quad (16)$$

Here Γ^c is the TEU coefficient and p^t is the number of ports visited by the transport. This favours the selection of containers for long-haul transport. Constraint (2) ensures that the vessel is seaworthy, and fulfils all the limits outlined in the vessel profile.¹ Constraint (3) links the weight variable w_{bp} with the weight of the containers stowed. Equations (4)-(8) are the block capacity constraints. For each block of the vessel we distinguish between a TEU and a 20-foot capacity. This

¹For a complete description of the polyhedron \mathcal{W} see Appendix A.

is due to the layout of the vessel since not all the blocks can stow 20-foot containers. In most of the cases k_b^{20} is, however, equal to k_b^{TEU} . Constraints (4) and (5) ensure that both of these capacities are satisfied. Similar to the TEU capacity constraints, the reefer constraints are split into a slot capacity constraint and a cell capacity constraint. Each reefer container needs to be plugged into an electrical output to get power for the cooling unit and r_b^{Slot} describes the number of such slots in a block b . A reefer cell is a cell where one of the slots has a reefer plug, and this number is denoted by r_b^{Cell} , if both slots in a reefer cell have a reefer plug, then $r_b^{Slot} = 2r_b^{Cell}$. In more general terms $r_b^{Cell} \leq r_b^{Slot} \leq 2r_b^{Cell}$ will be satisfied for all blocks. Constraint (6) restricts the number of reefer slots that can be used, and (7) is the reefer cell capacity constraint. Here we multiply with $\frac{1}{2}\Gamma^c$ as a 40-foot container occupies a full cell, whereas a 20-foot container occupies half a cell, and due to (6) the slot capacity will be satisfied. Constraint (8) limits the total volume of the containers in a block. This is needed in order to account for the fact that high-cube containers are higher and thus use more than a simple TEU capacity.

Constraints (9) and (10) are the block assignment constraints and together they enforce that no overstowage can occur. Specifically, constraint (9) ensures that exactly one port is assigned as the discharge port for every port. Constraint (10) makes sure that containers from a transport t can only be stowed in a block if the assigned destination matches the transports destination, d^t , during the full journey. Constraint (9) does not enforce that if $\sigma_{bp'}^d = 1$ for a block b then all the ports until d' must have d' as discharge port. This will, however, be the case anyway since constraint (10) is posted for every port, which effectively disallows the stowage of containers with another ports of destination.

Constraint (11) ensures that no more containers than the ones expected in the cargo-flow can be selected for stowage. Finally (12)-(14) defines the variables domain, and (12) enforces the block weight capacity limit. The model includes ballast water within the set \mathcal{W} , but only as a way to fix stability issues, thus the ballast water is not minimised as part of the objective as in Delgado (2013).

The block stowage requirement is handled similar to how Ambrosino et al. (2015a,b) handle it. The main difference is that we strictly enforce that overstowage cannot happen, where it is part of the objective in the MP-MBPP described in Ambrosino et al. (2015a,b).

5 Solution Method

The model presented in Section 4 is in practice intractable to solve using standard MIP solvers (As shown by Table 1). Having integer and binary variables most commonly means that extra effort is needed for a MIP solver, and thus the σ_{bp}^d variables have a large impact on the effectiveness of the MIP model. Moreover, big-M constraints, as the ones in (10) deteriorate the LP-bound. The basic idea of the forthcoming matheuristic is to use a heuristic to fix the block assignment and effectively remove the σ_{bp}^d variables from the model. Additionally the equality requirement in (3) enforces a correspondence between the container weight variables and the total weight of a block. Such constraints often require extra effort from MIP solvers, thus the relaxation of these constraints could speed up the execution.

The proposed matheuristic is a three-phase procedure as illustrated in Figure 6. The algorithm takes as input a vessel profile, a vessel service, and an expected cargo flow and returns a feasible solution. It is a hierarchical algorithm, as the flow of information only goes downward, and the lower stages do not feed information back to the upper stages. Phase I heuristically determines the block assignments at each port. This then becomes a variable fixing of the σ_{bp}^d variables in the second phase. In Phase II the weight equality in (3) is relaxed, as is the integrality requirement of the cargo assignment variables (y_b^{tc}). The weight variables w_{bp} are used in the stability constraints. Therefore, to ensure the stability of the vessel w_{bp} must correspond to the actual weight stowed in a block. This exact correspondence is broken because of the relaxation of (3) in Phase II. The

solution might then result in an unstable vessel. This is corrected in Phase III which receives as input from Phase II the values of the y_b^{tc} variables and solves a problem where the y_b^{tc} values are used as an upper bound on the cargo load, while also ensuring the solution is feasible wrt. to stability and integrality requirements.

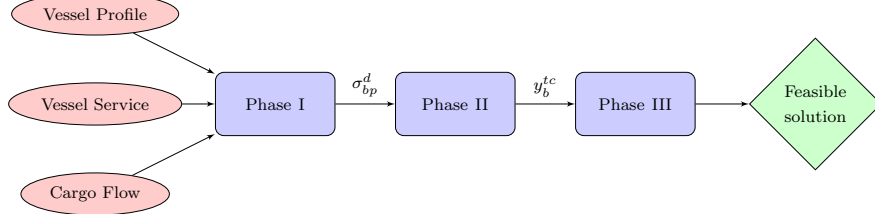


Figure 6: Flow chart outlining the flow of the matheuristic

5.1 Phase I

In the first phase, *schedules* are generated. A schedule s can be thoroughly described by the following parameter

$$b_{sp}^d = \begin{cases} 1, & \text{if schedule } s \text{ assigns destination } d \in \mathcal{P} \text{ as the discharge port when in port } p \in \mathcal{P} \\ 0, & \text{otherwise.} \end{cases}$$

A schedule s defines a block assignment for a full rotation, and is similar to the variables σ_{bp}^d , with the difference that σ_{bp}^d is for a given block. The first phase of the matheuristic must essentially do two things; generate schedules, and assign schedules to blocks.

When generating schedules, the goal is to find good schedules that fit the attributes of a specific type of block. We do so by considering the capacity resource and the possible assignments of container types. For every block, a schedule is generated by solving a longest path problem in a directed acyclic graph. Figure 7 is an example graph with 3 ports, the label on the nodes is the port symbolised by the node.

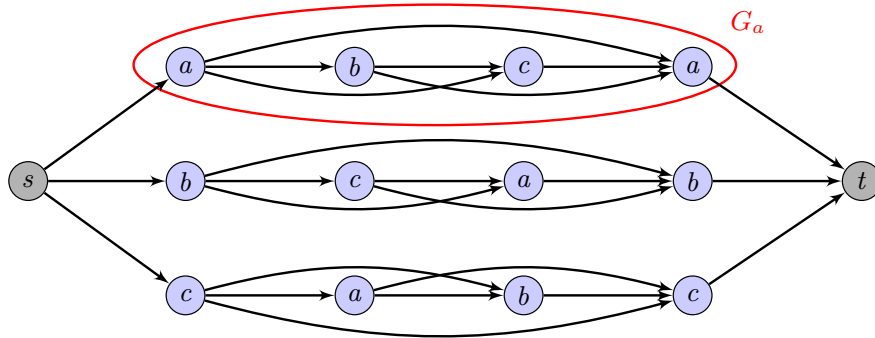


Figure 7: An example graph G of 3 ports, with the subgraph G_a highlighted.

An edge in the graph G describes a block assignment and a s - t path in G describes a schedule. If a path in G uses the edge (a, c) , the corresponding schedule assigns discharge port c at port a . At all the ports in between a and c this assignment must be respected, and thus c is assigned as the discharge port for all these ports as well. Therefore, for the path (s, a, b, a, t) in G the corresponding

schedule, s' , assigns b as the discharge port when in port a , and discharge port a is assigned for port b and c . In terms of the parameter $b_{s'a}^b$, this corresponds to the following

$$b_{s'a}^b = 1, \quad b_{s'b}^a = 1, \quad b_{s'c}^a = 1$$

Let G be a weighted directed acyclic graph with $V(G)$ as the vertex set and $E(G)$ as the set of edges. The graph G consists of a source s , a sink t , and $|P|$ mutually disjoint subgraphs, G_p , such that

$$V(G) \setminus \{s, t\} = \bigcup_{p \in P} V(G_p) \quad \wedge \quad \bigcap_{p \in P} V(G_p) = \emptyset$$

Each node, i , in G_p symbolises a port in the rotation, this correspondence is described by the function $p(i)$ (see Figure 7 where the subgraph G_a is highlighted). The subgraph $G_{p'}$ contains exactly $|P| + 1$ vertices. Let i_n be the n 'th vertex in a topological sorting of the vertices in $G_{p'}$, then

$$p(i_n) = \begin{cases} p', & n = 1 \\ p(i_{n-1}) + 1, & 2 \leq n \leq |P| + 1. \end{cases}$$

Where p' is a port in the rotation and $p' + 1$ is the port visited immediately after p' .

Let $t_p(i)$ be the position of the vertex i in a topological sorting of the vertices of G_p , then let $\delta_p^+(i)$ describe the set of vertices, j , where $t_p(i) < t_p(j)$. The edge set of G_p can now formally be defined as

$$\begin{aligned} \delta_p^+(i) &= \{j \in V(G_p) \mid t_p(i) < t_p(j)\} & \forall i \in V(G_p), p \in P \\ E(G_p) &= \{(i, j) \mid i \in V(G_p), j \in \delta_p^+(i)\} & \forall p \in P \end{aligned}$$

Along with the edges from the subgraphs, G contains edges that connects the subgraphs to the source and the sink. Let $E(G_{base})$ be this set of edges defined as

$$E(G_{base}) = \{(s, j) \mid \exists p : t_p(j) = 1\} \cup \{(i, t) \mid \exists p : t_p(i) = |P| + 1\}$$

With this the edge set of G can be properly defined as

$$E(G) \setminus E(G_{base}) = \bigcup_{p \in P} E(G_p)$$

The weight w_{ij} , for a given edge (i, j) , represents the maximum number of containers that can be transported on the edge. The weights are heuristically set following the procedure described in Algorithm 1.

Algorithm 1 takes as input graph G , demand D , value matrix F , the block TEU (τ) and reefer TEU capacity (τ^r). The algorithm returns the edge weight matrix W . For each block b the TEU and reefer capacities can be defined as follows:

$$\tau = \max(k_b^{20}, k_b^{TEU}), \quad \tau^r = \max(r_b^{Slot}, 2r_b^{Cell})$$

For the edges in $E(G_{base})$ the weight is 0, and therefore the loop in line 1 excludes these edges. Line 2-4 initialises the remaining TEU and reefer capacity for the edge (i, j) . For an edge (i, j) , the loop in lines 5 - 21 iterates over all the intermediate ports in the same order as they are visited, and calculate the weight of this edge. Here, $P_{p(i)}^{p(j)}$ is the set of ports visited between port $p(i)$ and port $p(j)$, including $p(i)$ but excluding $p(j)$. In line 7 the container types are sorted by value in descending order since we want to assign the most profitable containers first. For each container type c , we then have to calculate the available capacity. Lines 9-13 calculate the capacity of the

Algorithm 1 Calculate w_{ij}

Input: G, D, F, τ, τ^r ,
1: **for all** $(i, j) \in E(G) \setminus E(G_{base})$ **do**
2: $w_{ij} \leftarrow 0$
3: $\Delta\tau \leftarrow \tau$
4: $\Delta\tau^r \leftarrow \tau^r$
5: **for all** $k \in P_{p(i)}^{p(j)}$ **do**
6: $t \leftarrow$ The transport $\langle k, p(j) \rangle$
7: $sortC \leftarrow$ Container types of transport t sorted by descending order of f^{tc}
8: **for all** $c \in sortC$ **do**
9: **if** $c \in \mathcal{R}$ **then**
10: capacity $\leftarrow \min(\Delta\tau, \Delta\tau^r)$
11: **else**
12: capacity $\leftarrow \Delta\tau$
13: **end if**
14: load $\leftarrow \min(d_{kp(j)}^c, \lfloor \frac{capacity}{\Gamma^c} \rfloor)$
15: $w_{ij} \leftarrow w_{ij} + f^{tc}load$
16: $\Delta\tau \leftarrow \Delta\tau - \Gamma^cload$
17: **if** $c \in \mathcal{R}$ **then**
18: $\Delta\tau^r \leftarrow \Delta\tau^r - \Gamma^cload$
19: **end if**
20: **end for**
21: **end for**
22: **end for**
23: **return** W

specific type. If it is a reefer container we both need to account for the remaining reefer capacity, as well as the remaining TEU capacity, otherwise the remaining TEU capacity is sufficient. Line 14 determines the number of containers that can be assigned the edge. Here $d_{kp(j)}^c$ is the demand from port k to port $p(j)$ of container type c . The number of containers that can be assigned the edge is the minimum between the demand and the calculated capacity. We divide the capacity by Γ^c , the TEU coefficient, to have the actual number of containers. Line 15 updates the edge weight, while lines 16-19 update the remaining TEU capacity, $\Delta\tau$, and the remaining reefer capacity, $\Delta\tau^r$.

The edge weights are set in such a way that the longest path in the graph corresponds to a block assignment where the structure of the demand is taken into account. For instance, ports with a comparatively high number of unloading containers will be chosen as the destination for more origin ports. The longest path problem is solved for each block. The assigned cargo is then subtracted from the demand matrix to account for the cargo the generated schedule can carry. The edge weights calculations are based on the demand and thus the edge weights are updated for every block. Therefore, the longest path differs from block to block.

The graph G is a directed acyclic graph, and the longest path problem can be solved efficiently by topological sorting the vertexes and using a slightly modified version of Dijkstra's algorithm, such that the longest path is found instead of the shortest.

5.2 Phase II

In the second phase the weight equality in (3) is relaxed, and a mixed integer programming model assigns containers to blocks such that the block assignment is satisfied.

Let the parameter $\hat{\sigma}_{bp}^d$ describes the block assignment from Phase I as follows:

$$\hat{\sigma}_{bp}^d = \begin{cases} 1, & \text{If block } b \in \mathcal{B} \text{ is assigned destination } d \in \mathcal{P} \text{ at port } p \in \mathcal{P} \\ 0, & \text{Otherwise.} \end{cases}$$

Hence the interpretation of $\hat{\sigma}_{bp}^d$ is similar to that of the variable σ_{bp}^d . The integrality constraint of

y_b^{tc} are relaxed resulting in the following Phase II model:

$$\text{Max (1)}$$

Subject to:

$$\begin{aligned} & (2) \\ & (4) - (8) \\ & (11) - (12) \\ & \sum_{t \in \mathcal{T}_p^{ON}} \sum_{c \in \mathcal{C}} v^c y_b^{tc} \leq w_{bp} \quad \forall b \in \mathcal{B}, p \in \mathcal{P} \end{aligned} \quad (17)$$

$$y_b^{tc} \leq a^{tc} \widehat{\sigma}_{bp}^{dt} \quad \forall b \in \mathcal{B}, c \in \mathcal{C}, p \in \mathcal{P}, t \in \mathcal{T}_p^{ON} \quad (18)$$

$$y_b^{tc} \geq 0 \quad \forall b \in \mathcal{B}, t \in \mathcal{T}, c \in \mathcal{C} \quad (19)$$

Constraint (17) is a relaxation of the weight setting constraint (3), while (18) ensures that the block assignment from Phase I is satisfied. We can omit constraint (9) since it is implicitly satisfied by the fixed variable assignment $\widehat{\sigma}_{bp}^{dt}$. The y_b^{tc} variables are enforced to be non-negative in (19). Additionally, the integrality requirement is relaxed to speed up the execution of Phase II.²

Due to the relaxation of (3) and (19), this model might overestimate the number of loaded containers, as the weight variables used in the stability calculation is not strictly enforced to match the actual weight in the block.

Experimental evaluation has shown that this model can find high-quality solutions fast. Closing the final gap between the lower and upper bound and proving optimality is, however, time-consuming. For this reason, we impose an optimality tolerance η , meaning that the solution process is stopped once the gap between the lower and upper bound reaches this tolerance value.

5.3 Phase III

Phase III uses the solution from Phase II as a starting point and ensures that the final vessel configuration is stable, while enforcing the integrality requirement of the cargo assignment variables.

First let \widehat{y}_b^{tc} describe the container placement solution from Phase II, and define the variable $u_b^{tc} \in \mathbb{N}$ as the number of containers of type c in transport t to be stowed in block b . To decrease the computational effort needed to ensure the feasibility of the solution, the u_b^{tc} variables are restricted to be less than or equal to \widehat{y}_b^{tc} .

The model solved in Phase III is the following

$$\text{Min } Z = \sum_{b \in \mathcal{B}} \sum_{t \in \mathcal{T}} \sum_{c \in \mathcal{C}} f^{tc} \widehat{y}_b^{tc} - f^{tc} u_b^{tc} \quad (20)$$

Subject to:

$$\begin{aligned} & (2) \\ & (12) \\ & \sum_{t \in \mathcal{T}_p^{ON}} \sum_{c \in \mathcal{C}} v^c u_b^{tc} = w_{bp} \quad \forall b \in \mathcal{B}, p \in \mathcal{P} \end{aligned} \quad (21)$$

$$0 \leq u_b^{tc} \leq \widehat{y}_b^{tc} \quad \forall b \in \mathcal{B}, t \in \mathcal{T}, c \in \mathcal{C} \quad (22)$$

$$u_b^{tc} \in \mathbb{N} \quad \forall b \in \mathcal{B}, t \in \mathcal{T}, c \in \mathcal{C} \quad (23)$$

²Due to a binary indicator variable in the stability constraints, the problem is still a MIP.

The objective function (20) minimises the difference between the solution from Phase II and the final solution. This corresponds to removing the minimum number of containers, in terms of objective value. Constraint (21) sets the w_{bp} variable, forcing it to be equal to the weight stowed in block b at port p . Constraint (22) restricts the u_b^{tc} variables to be less than or equal to \widehat{y}_b^{tc} . The solution from Phase II respects the capacity constraints, and by only removing containers these constraints will still be satisfied. Therefore the capacity constraints can be disregarded in this phase. Finally (23) defines the domain for the cargo assignment variables.

6 Data Description

The vessel used for the experiments is operated by our industrial collaborator and has approximately 50 blocks and a TEU capacity of ca. 15,000 TEUs. The vessel data is confidential, thus only the vessel capacity can be made public.

Based on this vessel, we have generated a benchmark of 170 demand instances. All instances have been grouped by the number of ports to visit, which ranges between 4 and 20. Each group consists of 10 instances with varying demands. The ports are selected randomly from a list of 28 ports that are visited on the Europe-Asia services operated by our industrial collaborator. The order the ports are visited in is preserved, and the port-to-port demand and port drafts are based on data from LinerLib (Brouer et al., 2014). The demand from the LinerLib does not include transhipments (as those are part of the network design problem), however, it can be used as indication. To make sure that the demand corresponds to a cargo-flow which tends to utilize the full vessel capacity, we multiply the demand of each service leg by a scalar $\gamma = \text{number of ports} \cdot \text{nominal intake} / \text{LinerLib total demand}$. Successively the demand of each leg is decomposed into a demand for each container type by predefined probabilities. The probability for a container being a forty foot container is 40%, the reefer probability is 10%, and high-cube 10%. These probabilities are based on numbers from our industrial collaborator. The probabilities for a container having weight class w is denoted $P(w)$ and a reasonable distribution could be the following.

$$P(3) = 5\%, \quad P(6) = 10\%, \quad P(9) = 15\%, \quad P(14) = 25\%, \quad P(21) = 25\%, \quad P(27) = 20\%$$

These probabilities are sufficient to give a full probability distribution for all container types. The probability for generating a 40 foot, high-cube non-reefer container of weight 21 is, for example,

$$0.40 \cdot 0.10 \cdot (1 - 0.10) \cdot 0.25 = 0.9\%$$

Finally, using the value of a dry 40-foot container from the LinerLib, we calculate the revenue of each container type as described in Delgado (2013).

7 Computational Results

Four different solution methods have been tested and compared.

- **CMPBSMIP - CMPBS Mixed Integer Programming model**
The mixed integer programming model as presented in Section 4.
- **CMPBSHLP - CMPBS Heuristic Linear**
The 3-phase matheuristic proposed in Section 5.
- **CMPBSHIP - CMPBS Heuristic Integer**
This is a version of the 3-phase matheuristic where we do not relax the integrality constraints on the y_b^{tc} variables of Phase II.

- **2-CMPBSH - 2 Phased CMPBS Heuristic**

This is a version of the matheuristic where both the integrality constraints of the y_b^{tc} variables and the equality of constraint (3) are not relaxed. In this specific case, Phase III is no longer needed.

All methods have been implemented in JAVA 1.7 and are tested using a 2.30 GHz Intel Xeon E5 processor. CPLEX v. 12.6.1 is used to solve the MIP models in Section 4, Section 5.2 and Section 5.3 and each model is solved using 8 threads. For CMPBSMIP, a time limit of 1 hour has been used. For the three heuristics, the time limit is 10 minutes. For all methods, an optimality tolerance $\eta = 0.05\%$ was used.

Since only very few optimal solutions of the CMPBS can be calculated, we propose an upper bound to estimate the quality of the solution methods. The upper bound was computed by removing the block stowage requirement from the cargo mix model (constraints (9)-(10)). The best dual bound after 30 minutes of solving time is used as the upper bound for the CMPBS, and it is referred to as UB in the following.

7.1 Intake optimisation

For the intake optimisation, Table 1 shows aggregated results for the 4 tested method. Here $|P|$ describes the number of ports. #Sol is the number of instances where a feasible solution is found, the \bar{x} columns are the average solution value, lastly \bar{t} is the average execution time in seconds. The solution values reported for CMPBSMIP are from the best feasible solutions found within the time limit. Thus the values are not ensured to be better than that of the heuristics.

Table 1 shows that the CMPBSMIP model is too complex to solve real world sized instances. For instances with more than 6 ports the quality of the solutions are too bad, and for instances, with 11 or more ports the MIP model fails to even find a feasible solution within an hour. In total, only 2 instances are solved to optimality, both instances only having 4 ports. CMPBSHLP finds good quality solutions in just 14 seconds. Comparing with CMPBSHIP it can be seen that the relaxation of the cargo variables y_b^{tc} does not have a significant impact on solution quality, but the computational times are increased considerably. The best solutions are found with 2-CMPBSH. The method, however, is 8 times slower than CMPBSHLP. All three heuristic methods find feasible solutions for all the tested instances.

$ P $	UB	CMPBSMIP		CMPBSHLP		CMPBSHIP		2-CMPBSH		
	\bar{x}	#Sol	\bar{x}	\bar{t}	\bar{x}	\bar{t}	\bar{x}	\bar{t}	\bar{x}	\bar{t}
4	53265	10	46352	2957.5	50351	1.3	50551	5.6	50851	4.5
5	68153	8	58514	3600.0	62917	2.1	63090	5.5	63625	6.6
6	85935	5	41543	3600.0	81209	2.0	81472	9.4	81834	17.7
7	102409	4	39484	3600.0	93191	3.9	93490	20.2	94387	27.5
8	122125	7	48448	3600.0	112787	5.2	113198	19.9	113339	23.4
9	131360	1	53942	3600.0	122022	4.4	122375	28.9	122742	38.2
10	149285	3	54073	3600.0	138485	7.5	138996	28.1	139113	55.6
11	170912	0	-	3600.0	155353	7.4	155884	48.0	155967	61.8
12	185836	0	-	3600.0	168760	17.4	169457	63.4	169473	88.9
13	201501	0	-	3600.0	185299	17.4	186007	129.6	186024	100.6
14	212244	0	-	3600.0	196018	13.2	196596	66.5	197172	97.3
15	231770	0	-	3600.0	212763	18.9	213544	82.8	213546	154.3
16	249557	0	-	3600.0	229117	23.2	229921	120.5	229933	187.0
17	265357	0	-	3600.0	245174	19.4	245936	135.0	246106	239.1
18	280448	0	-	3600.0	258001	43.2	258989	201.3	259060	238.2
19	296581	0	-	3600.0	273704	25.7	274680	171.7	274689	246.6
20	308392	0	-	3600.0	285586	24.6	286621	125.5	286935	281.7
Average	183243		48908	3562.2	168867	13.9	169459	74.2	169694	109.9

Table 1: Intake optimisation: Aggregated results

For the 10 instances with 4 ports, Table 2 compares the upper bound with the solution and the bound after 5 hours of solving the model together with the CMPBSHLP. The '*' symbol denotes instances where the model execution was terminated due to the optimality tolerance $\eta = 0.05\%$. A time limit of 1 hour or 5 hours does not change the number of optimal solutions found, but for the non-optimal instances, the solution quality is improved. For all the instances with 4 ports, when given the CMPBSMIP 5 hours instead of 1 hour, the model finds better solutions and outperforms the CMPBSHLP heuristic with respect to solution quality. For the two optimal solutions, the table shows that the upper bound does not differ much from the optimal solution. For the rest of the instances, the upper bound is comparable to the bound achieved after 5 hours of solving the compact model. This indicates that the objective value found by the upper bound method is close to the optimal solution.

P	i	UB	CMPBSMIP			CMPBSHLP	
		x	x	x_{bound}	t	x	t
4	0	41894	41742	41893	18000	39014	1.1
4	1	54016	53992*	54017	690	51898	0.8
4	2	59349	57963	59361	18000	55012	2.6
4	3	59314	59202	59314	18000	57387	1.4
4	4	57242	57081	57225	18000	53713	1.0
4	5	45185	45165*	45186	311	44339	1.2
4	6	56634	56046	56632	18000	54556	1.7
4	7	47172	47115	47172	18000	45833	1.1
4	8	60146	59260	59832	18000	57001	1.1
4	9	51693	51062	51693	18000	44756	1.4
Average		53265	52863	53233	14500	50351	1.3

Table 2: Upper bound analysis

Table 3 evaluates the quality of the schedules obtained from solving the longest path problem in Phase I. The graph based method is compared with a random method, where schedules are generated at random. The table reports the final solution after Phase III. The solution in the Graph column is identical to the CMPBSHLP results from Table 1. To account for the randomness of the Random method, each instance is solved 10 times with a different set of random schedules. \bar{x}^b reports the average of the best solutions for the considered instance group, and \bar{x} is the average of all solutions for the instance group. The table shows that the graph based method substantially outperforms the random method. This is also to be expected as the graph based method cleverly generates schedules based on the demand.

Figure 8 shows an analysis of how different factors impact the ship occupancy. Solutions to the CMPBS tend to load a high number of light containers (as further described in Section 7.2). Figure 8a illustrates the occupancy rate of the vessel when the CMPBS is allowed to only load containers of a certain weight class. It can be easily seen that when we force the model to use the heavier containers, the ship occupancy drops resulting in a smaller intake.

In a similar fashion, the graph in Figure 8b illustrates the impact of draft restrictions at port. The graph shows the average ship occupancy over all instances, where the draft has been fixed to a single value for all ports. A small draft restricts the total displacement of the vessel, which means fewer containers can be loaded. When the draft is 15.5m or above the impact of the draft constraint is minor, and the ship occupancy is nearly constant.

It is worth noticing that the use of ballast tanks, not only helps satisfy the vessel seaworthiness requirements, they also impact the solution time of the method (Pacino et al., 2012). The variables that refer to the ballast tanks (see Appendix A) are continuous and behave as slack variables on the stability constraints. We show this in Figure 8c which depicts both the impact with respect to the ship occupancy and the solution time for the CMPBSHLP method. Each of the ballast tanks has a maximum capacity, this capacity is multiplied by a number between 0 and 1, to analyse its

$ P $	UB		Graph		Random	
	\bar{x}	\bar{x}	\bar{x}	\bar{x}	\bar{x}^b	\bar{x}^b
4	53265	50351	33383	37037		
5	68153	62917	37632	42985		
6	85935	81209	49896	55470		
7	102409	93191	56833	61875		
8	122125	112787	70806	78137		
9	131360	122022	72862	81859		
10	149285	138485	81115	90149		
11	170912	155353	96576	105861		
12	185836	168760	109886	120247		
13	201501	185299	112737	124068		
14	212244	196018	114017	126599		
15	231770	212763	124640	139470		
16	249557	229117	137357	151468		
17	265357	245174	141310	158254		
18	280448	258001	155261	169868		
19	296581	273704	166183	182065		
20	308392	285586	162003	179398		
Average	183243	168867	101323	112048		

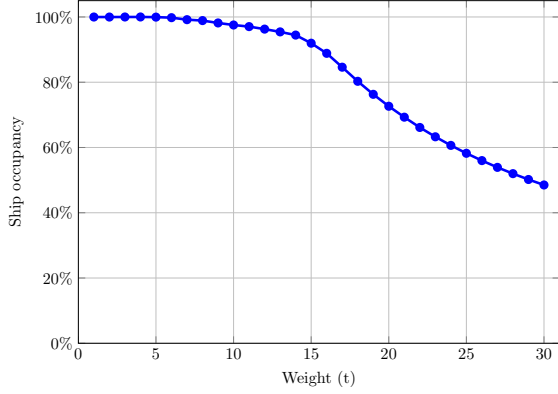
Table 3: Analysis of the quality of the schedules from Phase I.

impact. The graph shows that the ship occupancy slightly increases the bigger the capacity. This is expected as the more water that can be loaded into the ballast tanks the easier it is to satisfy stability issues. Also with respect to solution time the results are as expected. The graph shows that the higher the tank capacity, the less time is needed to solve the problem. The impact on the runtime of Phase II is minor. For low tank capacities, more time is spent in Phase III where the stability constraints are strictly enforced.

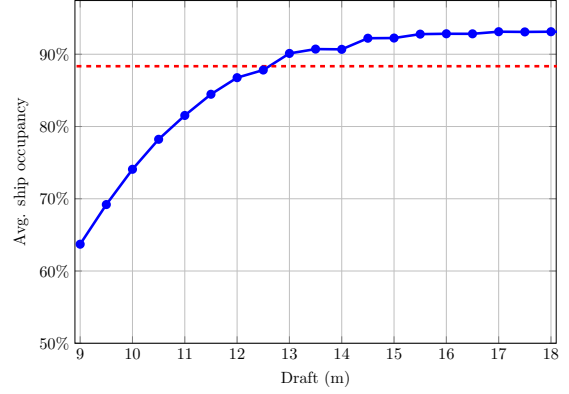
7.2 Revenue Optimisation

Table 4 shows aggregated results for the 4 tested methods when optimising revenue, instead of intake. The \bar{x} columns show the average revenue in millions of USD, the rest of the columns are similar to those of Table 1 described in Section 7.1. As can be seen, the solution methods behave, in terms of solution quality and runtime performance, in the same way as for intake optimisation. The same conclusions can thus be drawn.

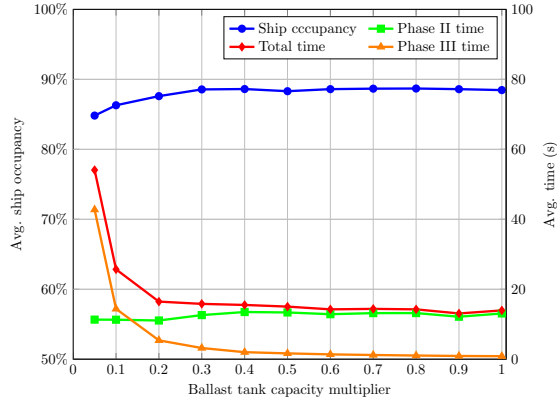
The difference between intake and revenue optimisation is clearer if we look at the type of cargo that is being loaded. Figure 9 shows a histogram of the containers occupying the vessel during the full rotation. The data is based on the average over all 170 instances. Figure 9a and Figure 9b is for the case of intake optimisation, and Figure 9c and Figure 9d is for the case of revenue optimisation. Figure 9a and Figure 9c show the container types loaded, here the label corresponds to the type of container, first the length, then reefer types are denoted with a R, and high-cubes with HC. For intake and revenue optimisation the distribution of container types is similar. The loading of reefer containers is expected to be low, even in revenue optimisation, as it is constrained by the available reefer plugs in the vessel. High-cubes tend to reduce capacity due to the extra volume, so it also makes sense that not many of them are loaded in either intake or revenue optimisation. Figure 9b and Figure 9d show the weight distribution of the 20' and 40' containers types loaded (excluding reefer and high-cubes). When optimising the intake, the value of a container does not depend on the weight, and therefore the model has a tendency to load light and empty containers (3 tonnes). When loading light containers the weight capacity is of less importance, and thus more containers can be loaded in total. However, empty containers do not generate as much revenue, and therefore only a few empties are loaded when optimising revenue. Though heavier containers generate more revenue, they also reduce vessel capacity since the weight limits of the stacks are reached faster, as



(a) Maximum achievable ship occupancy when only containers of a certain weight class are available.



(b) Average ship occupancy, over all instances, when all ports have the same draft.



(c) Average ship occupancy, over all instances, when limiting the ballast water allowed.

Figure 8: Ship occupancy analysis

was shown in Figure 8a.

Finally in Table 5 we show the impact of the relaxation of the weight constraints in Phase II for the CMPBSHLP matheuristic. Here \bar{x} is the solution value after the corresponding phase, and \bar{t} is the average runtime of each phase. The column %Rem is the percentage of removed containers by Phase III in terms of objective value. In both cases, the most time is spent in Phase II, of which approximately 10% is spent in Phase III. As can be seen Phase III only removes in average ca. 0.75% of the containers in terms of objective values.

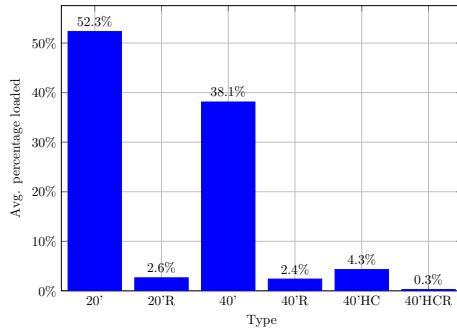
8 Conclusion

In this paper, the Cargo Mix Problem with Block Stowage has been introduced. The problem has been described thoroughly and formulated as a mathematical model. The mathematical model is not scalable, and cannot solve any of the real-life sized instances. To overcome this, a matheuristic has been developed. The method combines a 3-phase decomposition with variable fixing and heuristic assignments.

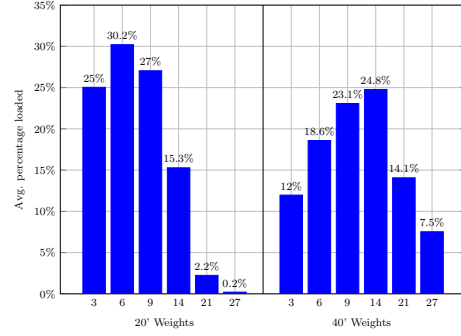
Three variations of the matheuristic were tested on 170 data instances, both when optimising revenue and intake. The results showed that the approach finds high-quality solutions in seconds and can scale to industrial size instances.

P	UB	CMPBSMIP			CMPBSHLP		CMPBSHIP		2-CMPBSH	
	\bar{x} (10 ⁶ \$)	#Sol	\bar{x} (10 ⁶ \$)	\bar{t}	\bar{x} (10 ⁶ \$)	\bar{t}	\bar{x} (10 ⁶ \$)	\bar{t}	\bar{x} (10 ⁶ \$)	\bar{t}
4	29.93	10	29.77	2512.2	28.70	0.9	28.81	2.8	28.95	6.3
5	37.13	10	35.27	3007.2	35.88	1.3	36.00	4.9	36.03	7.5
6	36.99	9	26.76	3600.0	35.15	2.3	35.30	10.1	35.33	18.8
7	40.89	7	16.90	3600.0	39.07	2.9	39.22	20.5	39.37	39.9
8	41.79	7	13.65	3600.0	39.52	4.6	39.69	26.0	39.75	39.7
9	42.68	2	6.08	3600.0	40.54	3.6	40.71	27.6	40.92	65.0
10	43.10	1	1.96	3600.0	40.67	4.3	40.83	20.9	40.96	87.8
11	47.03	1	2.82	3600.0	44.19	10.0	44.39	47.6	44.35	134.9
12	48.19	0	-	3600.0	45.27	9.5	45.46	62.9	45.50	123.9
13	49.67	0	-	3600.0	46.65	13.6	46.84	59.1	47.08	183.4
14	50.50	0	-	3600.0	47.56	8.5	47.73	98.3	47.21	387.0
15	49.40	0	-	3600.0	46.57	17.1	46.73	91.6	46.94	170.7
16	50.26	0	-	3600.0	47.53	20.2	47.73	132.6	47.72	407.0
17	51.93	0	-	3600.0	48.16	18.4	48.34	103.9	48.61	218.5
18	53.98	0	-	3600.0	49.81	25.7	50.00	94.4	50.10	268.5
19	55.51	0	-	3600.0	51.47	33.5	51.68	163.2	49.80	370.7
20	51.83	0	-	3600.0	48.05	27.6	48.22	193.4	48.55	318.2
Average	45.93		16.65	3501.1	43.22	12.0	43.39	68.2	43.36	167.5

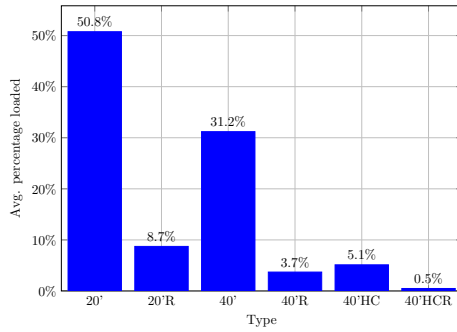
Table 4: Revenue optimisation: Aggregated results



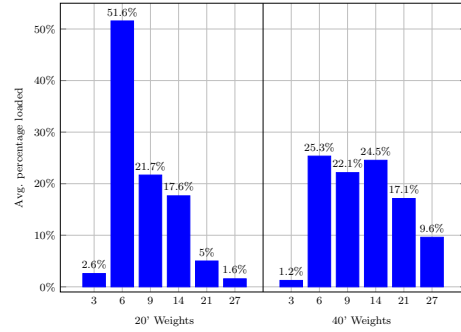
(a) Intake optimisation: Container load distribution



(b) Intake optimisation: Container weight load distribution



(c) Revenue optimisation: Container load distribution



(d) Revenue optimisation: Container weight load distribution

Figure 9: Containers loaded when optimising the intake and the revenue.

Though the described problem can be seen as a strategic planning tool, its analytical strength is better suited at the operational level e.g. in uptake management. Since the solution approach is based on mathematical models, it is easy to add extra constraints to e.g. perform what-if scenario

P	Intake					Revenue				
	Phase II		Phase III			Phase II		Phase III		
	\bar{x}	\bar{t}	\bar{x}	\bar{t}	%Rem	\bar{x} (10^6 \$)	\bar{t}	\bar{x} (10^6 \$)	\bar{t}	%Rem
4	50985	1.1	50351	0.3	1.35%	29.00	0.7	28.70	0.2	1.03%
5	63883	1.8	62917	0.3	1.53%	36.04	1.0	35.88	0.2	0.44%
6	82100	1.6	81209	0.4	1.18%	35.36	1.9	35.15	0.4	0.56%
7	94995	3.4	93191	0.5	2.03%	39.49	2.3	39.07	0.6	0.97%
8	113379	4.5	112787	0.7	0.53%	39.77	4.0	39.52	0.6	0.61%
9	122824	3.7	122022	0.7	0.66%	40.96	2.7	40.54	0.9	1.00%
10	139182	6.7	138485	0.8	0.50%	41.01	3.3	40.67	1.0	0.77%
11	156014	6.7	155353	0.7	0.42%	44.43	9.3	44.19	0.7	0.50%
12	169533	16.6	168760	0.9	0.46%	45.52	8.5	45.27	1.0	0.50%
13	186085	16.4	185299	1.0	0.42%	47.26	12.1	46.65	1.4	1.21%
14	197610	12.1	196018	1.1	0.85%	47.94	7.0	47.56	1.5	0.76%
15	213619	18.0	212763	0.9	0.40%	47.02	15.4	46.57	1.7	0.92%
16	230005	22.0	229117	1.2	0.39%	47.84	18.3	47.53	1.9	0.60%
17	246212	17.9	245174	1.5	0.42%	48.66	16.1	48.16	2.3	0.96%
18	259156	41.8	258001	1.4	0.45%	50.24	23.0	49.81	2.7	0.80%
19	274787	24.3	273704	1.4	0.40%	51.78	31.6	51.47	1.9	0.57%
20	287183	23.0	285586	1.5	0.57%	48.63	24.5	48.05	3.1	1.09%
Average	169856	13.0	168867	0.9	0.74%	43.59	10.7	43.22	1.3	0.78%

Table 5: Detailed results for CMPBSHLP when optimising intake and revenue

analysis. To give an example, given the current state of a vessel, meaning its current load and remaining capacity, it is possible to use the CMPBS to identify an optimised number of containers to choose from a list of bookings. The impact of a booking could also be compared to an optimised cargo-mix to estimate the impact it will have on revenue. The authors believe that the mathematical models and the solution approach could easily be adapted to perform this kind of analysis within, for example, a decision support system. More so since the presented method can solve the CMPBS within seconds.

Possible areas of future research can focus on both optimal and heuristic methods. The decomposition currently adopted could be suitable for the implementation of a column generation approach where the schedules are dynamically generated. Further research, however, needs to be done for such an approach to be successful due to the complexity of the resulting reduced model.

Since the quality of the final solution heavily depends on the block assignment in Phase I, an idea for future research is to improve the way it is computed. This can be done by improving the way schedules are generated, and thus keeping the hierarchical nature of the algorithm, or by using an iterative procedure. In an iterative approach, the solution from one of the later phases is fed back to a previous phase to re-optimize, until some termination criterion is met. Within a decision support system setting, this approach is, however, not suitable for the current hierarchical heuristic since it would use an average of ca. 14 seconds for each iteration.

Furthermore, we plan to extend the problem by considering stochastic cargo flows, and we also plan to study how this work can be combined with other important problems in the liner shipping industry, e.g. service network design and cargo flow optimisation.

Acknowledgements

This work has been funded by the Danish Innovationsfonden under the GREENSHIP Project (1313-00005B-GREENSHIP).

References

- Ambrosino, D., Anghinolfi, D., Paolucci, M., and Sciomachen, A. (2009). A new three-step heuristic for the master bay plan problem. *Maritime Econ Logistics*, 11(1):98–120.
- Ambrosino, D., Anghinolfi, D., Paolucci, M., and Sciomachen, A. (2010). An experimental comparison of different heuristics for the master bay plan problem. In Festa, P., editor, *Experimental Algorithms*, volume 6049 of *Lecture Notes in Computer Science*, pages 314–325. Springer Berlin Heidelberg.
- Ambrosino, D., Paolucci, M., and Sciomachen, A. (2015a). Computational evaluation of a MIP model for multi-port stowage planning problems. *Soft Computing*, pages 1–11.
- Ambrosino, D., Paolucci, M., and Sciomachen, A. (2015b). A MIP heuristic for multi port stowage planning. *Transportation Research Procedia*, 10:725 – 734. 18th Euro Working Group on Transportation, EWGT 2015, 14-16 July 2015, Delft, The Netherlands.
- Ambrosino, D. and Sciomachen, A. (1998). A Constraint Satisfaction Approach for Master Bay Plans. *Water studies series*, 36.
- Ambrosino, D., Sciomachen, A., and Tanfani, E. (2004). Stowing a containership: the master bay plan problem. *Transportation Research Part A: Policy and Practice*, 38(2):81–99.
- Avriel, M., Penn, M., Shpirer, N., and Witteboon, S. (1998). Stowage planning for container ships to reduce the number of shifts. *Annals of Operations Research*, 76(1-4):55–71.
- Botter, R. C. and Brinati, M. A. (1992). Stowage container planning: A model for getting an optimal solution. In *Proceedings of the IFIP TC5/WG5.6 Seventh International Conference on Computer Applications in the Automation of Shipyard Operation and Ship Design, VII*, pages 217–229, Amsterdam, The Netherlands, The Netherlands. North-Holland Publishing Co.
- Brouer, B. D., Alvarez, J. F., Plum, C. E. M., Pisinger, D., and Sigurd, M. M. (2014). A Base Integer Programming Model and Benchmark Suite for Liner-Shipping Network Design. *Transportation Science*, 48:281–312.
- Christiansen, M., Fagerholt, K., Nygreen, B., and Ronen, D. (2007). Chapter 4 Maritime Transportation. In *Handbooks in Operations Research and Management Science*.
- Delgado, A. (2013). Models and Algorithms for Container Vessel Stowage Optimization.
- Delgado, A., Jensen, R., and Janstrup, K. (2012). A constraint programming model for fast optimal stowage of container vessel bays. *European Journal of Operational Research*, 220(1):251–261.
- Ding, D. and Chou, M. C. (2015). Stowage Planning for Container Ships: A Heuristic Algorithm to Reduce the Number of Shifts. *European Journal of Operational Research*.
- Dubrovsky, O., Levitin, G., and Penn, M. (2002). A genetic algorithm with a compact solution encoding for the container ship stowage problem. *Journal of Heuristics*, pages 585–599.
- Feng, C.-M. and Chang, C.-H. (2008). Optimal slot allocation in intra-asia service for liner shipping companies. *Maritime Economics & Logistics*, 10(3):295–309.
- Kang, J.-G. and Kim, Y.-D. (2002). Stowage planning in maritime container transportation. *Journal of the Operational Research Society*, 53(4):415–426.

- Li, F., Tian, C., Cao, R., and Ding, W. (2008). An integer linear programming for container stowage problem. *Computational Science ICCS 2008*, pages 853–862.
- Pacino, D., Delgado, A., Jensen, R., and Bebbington, T. (2012). An accurate model for seaworthy container vessel stowage planning with ballast tanks. In Hu, H., Shi, X., Stahlbock, R., and Vo, S., editors, *Computational Logistics*, volume 7555 of *Lecture Notes in Computer Science*, pages 17–32. Springer Berlin Heidelberg.
- Pacino, D., Delgado, A., Jensen, R. M., and Bebbington, T. (2011). Fast generation of near-optimal plans for eco-efficient stowage of large container vessels. In Böse, J. W., Hu, H., Jahn, C., Shi, X., Stahlbock, R., and Voß, S., editors, *Computational Logistics*, volume 6971 of *Lecture Notes in Computer Science*, pages 286–301. Springer Berlin Heidelberg.
- Pacino, D. and Jensen, R. (2012). Constraint-based local search for container stowage slot planning. *Lecture Notes in Engineering and Computer Science*, 2:1467–1472.
- Reinhardt, L. B. and Pisinger, D. (2012). A branch and cut algorithm for the container shipping network design problem. *Flexible Services and Manufacturing Journal*, 24:349–374.
- Sciomachen, A. and Tanfani, E. (2003). The master bay plan problem: a solution method based on its connection to the threedimensional bin packing problem. *IMA Journal of Management Mathematics*, 14:251–269.
- Wilson, I. and Roach, P. (2000). Container stowage planning: A methodology for generating computerised solutions. *The Journal of the Operational Research Society*, 51(11):1248–1255.
- Zurheide, S. and Fischer, K. (2011). *A Revenue Management Slot Allocation Model with Prioritization for the Liner Shipping Industry*, pages 143–148. Springer Berlin Heidelberg, Berlin, Heidelberg.

A Stability Constraints

The stability constraints ensure that the vessel does not capsize or break, even in tough weather conditions. The constraints here uses similar notation as the one introduced in Delgado (2013), and the interested reader is invited to look at the source for a more elaborate description.

The main difference between the set of constraints presented below and (6.25) - (6.43) of Delgado (2013), is the draft. Delgado (2013) does not consider draft limit for each port, thus making it possible for the vessel to have a greater displacement. The introduction of the LinerLib (Brouer et al., 2014) makes this data easily available and the draft limit is therefore included in this work. Also, Delgado considers locations, where this work considers blocks.

The functions describing the metacentre, trim, draft and buoyancy force all depends non-linearly on the displacement. However Delgado (2013) shows that these functions can be approximated by linear planes by splitting the full displacement range into *displacement intervals*. A displacement interval is thus defined as a minimum, maximum (W_i^- and W_i^+) and an average (W_i) weight for the interval.

The sets, variables and parameters used in the model are introduced and explained below.

Sets:

\mathcal{P}	Set of ports.
\mathcal{T}	Set of ballast tanks.
\mathcal{B}	Set of blocks.
\mathcal{I}	Set of displacement intervals.
\mathcal{BS}	Set of bonjean stations.
\mathcal{F}	Set of frames.

Decision Variables:

w_{bp}	$\in \mathbb{R}^+$	Weight stowed in block $b \in \mathcal{B}$ at port $p \in \mathcal{P}$.
x_{tp}	$\in \mathbb{R}^+$	Weight of the tank $t \in \mathcal{T}$ at port $p \in \mathcal{P}$.

Auxiliary Variables:

v_p^W	$\in \mathbb{R}^+$	Vessel Displacement at port $p \in \mathcal{P}$.
v_{ip}^W	$\in \mathbb{R}^+$	Vessel Displacement in interval $i \in \mathcal{I}$ at port $p \in \mathcal{P}$.
v_{ip}^L	$\in \mathbb{R}$	Longitudinal centre of gravity at displacement interval $i \in \mathcal{I}$ at port $p \in \mathcal{P}$.
ψ_{ip}	$\in \{0, 1\}$	Indicator variable for displacement interval $i \in \mathcal{I}$ at port $p \in \mathcal{P}$.
v_p^V	$\in \mathbb{R}^+$	Vertical centre of gravity at port $p \in \mathcal{P}$.
v_p^{VM}	$\in \mathbb{R}^+$	Vertical moment at port $p \in \mathcal{P}$.
v_p^M	$\in \mathbb{R}^+$	Metacentre at port $p \in \mathcal{P}$.
v_{bsp}^{Bs}	$\in \mathbb{R}^+$	Buoyancy force of section between bonjean station bs and $bs + 1$ at port $p \in \mathcal{P}$.
$v_{f\alpha p}^S$	$\in \mathbb{R}^+$	Shear force fore or aft of frame $f \in \mathcal{F}$ at port $p \in \mathcal{P}$.
$v_{f\alpha p}^B$	$\in \mathbb{R}^+$	Bending moment fore or aft of frame $f \in \mathcal{F}$ at port $p \in \mathcal{P}$.

Parameters:

Various parameters:

W^O	Weight of the empty vessel.
Min^{GM}	Lower bound for metacentric height
Max_p^D	Maximum draft allowed at port p

Displacement intervals parameters:

$W_i^{\{-,+\}}$	Lower (-) and upper (+) bound of displacement interval $i \in \mathcal{I}$
W_i	Average weight of displacement interval $i \in \mathcal{I}$
$A_{\{M,T,D,Bs\}}^W(W_i)$	Weight coefficient of displacement interval $i \in \mathcal{I}$ for the linearization of metacentre (M), trim (T), draft (D), and bonjean at station bs (Bs).
$A_{\{M,T,Bs\}}^L(W_i)$	Lcg coefficient of displacement interval $i \in \mathcal{I}$ for the linearization of metacentre (M), trim (T), and bonjean at station bs (Bs).
$A_{\{M,T,D,Bs\}}(W_i)$	Constant of displacement interval $i \in \mathcal{I}$ for the linearization of metacentre (M), trim (T), draft (D), and bonjean at station bs (Bs).

Centre of gravity parameters:

$\text{Min}_i^L/\text{Max}_i^L$	Minimum/maximum longitudinal centre of gravity at displacement interval $i \in \mathcal{I}$
$D_b^{\{L,V\}}$	Longitudinal (L), and Vertical (V) centre of gravity of block $b \in \mathcal{B}$
$D_t^{\{L,V,T\}}$	Longitudinal (L), Vertical (V), and transversal (T) centre of gravity of ballast tank $t \in \mathcal{T}$
Max^V	Maximum vertical moment possible for the vessel.
LM^O	Longitudinal moment of the empty vessel including constant weights.
VM^O	Vertical moment of the empty vessel including constant weights.
TM^O	Transversal moment of the empty vessel including constant weights.

Bending/Shearing parameters:

$W_{f\alpha}^S$	Constant weights fore or aft of frame $f \in \mathcal{F}$
$G_{\{b,t,bs\}f}^\alpha$	Fraction of block b , ballast tank t , and buoyancy section between bonjean stations bs and $bs + 1$ that lies fore or aft frame f
$W_{f\alpha}^B$	Bending components of the constant weight fore or aft of frame $f \in \mathcal{F}$
D_b^{Bs}	Distance in meters between bonjean stations bs and $bs + 1$ multiplied by the density of water
$A_{\{b,t,bs\}f}^\alpha$	Fore or aft distance from frame f to the longitudinal centre of gravity of block b , ballast tank t , buoyancy section between bonjean stations bs and $bs + 1$
G_f	Fore-based fraction of frame $f \in \mathcal{F}$, where $G_f \in [0; 1]$. $G_f = 1$ when f is the first frame at the bow, and $G_f = 0$ when f is the first frame at the stern.
$\text{Min}_f^{\{S,B\}}$	Lower bound for shear force (S) and bending moment (B) at frame $f \in \mathcal{F}$
$\text{Max}_f^{\{S,B\}}$	Upper bound for shear force (S) and bending moment (B) at frame $f \in \mathcal{F}$

With these the stability constraints can be modelled as seen below.

$$\sum_{t \in \mathcal{T}} x_{tp} + \sum_{b \in \mathcal{B}} w_{bp} + W^O = v_p^W \quad \forall p \in \mathcal{P} \quad (24)$$

$$\sum_{i \in \mathcal{I}} W_i^- \psi_{ip} \leq v_p^W \leq \sum_{i \in \mathcal{I}} W_i^+ \psi_{ip} \quad \forall p \in \mathcal{P} \quad (25)$$

$$\sum_{i \in \mathcal{I}} \psi_{ip} = 1 \quad \forall p \in \mathcal{P} \quad (26)$$

$$\sum_{i \in \mathcal{I}} v_{ip}^W = v_p^W \quad \forall p \in \mathcal{P} \quad (27)$$

$$W_i^- \psi_{ip} \leq v_{ip}^W \leq W_i^+ \psi_{ip} \quad \forall i \in \mathcal{I}, p \in \mathcal{P} \quad (28)$$

$$\text{Min}_i^L \psi_{ip} \leq v_{ip}^L \leq \text{Max}_i^L \psi_{ip} \quad \forall i \in \mathcal{I}, p \in \mathcal{P} \quad (29)$$

$$\sum_{b \in \mathcal{B}} D_b^L w_{bp} + \sum_{t \in \mathcal{T}} D_t^L x_{tp} + LM^O = \sum_{i \in \mathcal{I}} W_i v_{ip}^L \quad \forall p \in \mathcal{P} \quad (30)$$

$$v_p^V W_i + (1 - \psi_{ip}) \text{Max}^V \geq v_p^{VM} \quad \forall i \in \mathcal{I}, p \in \mathcal{P} \quad (31)$$

$$v_p^V W_i - (1 - \psi_{ip}) \text{Max}^V \leq v_p^{VM} \quad \forall i \in \mathcal{I}, p \in \mathcal{P} \quad (32)$$

$$\sum_{b \in \mathcal{B}} D_b^V w_{bp} + \sum_{t \in \mathcal{T}} D_t^V x_{tp} + VM^O = v_p^{VM} \quad \forall p \in \mathcal{P} \quad (33)$$

$$\sum_{t \in \mathcal{T}} D_t^T x_{tp} + TM^O = 0 \quad \forall p \in \mathcal{P} \quad (34)$$

$$\sum_{i \in \mathcal{I}} A_M^W(W_i) v_{ip}^W + A_M^L(W_i) v_{ip}^L + A_M(W_i) \psi_{ip} = v_p^M \quad \forall p \in \mathcal{P} \quad (35)$$

$$v_p^M - v_p^V \geq \text{Min}^{GM} \quad \forall p \in \mathcal{P} \quad (36)$$

$$\sum_{i \in \mathcal{I}} A_T^W(W_i) v_{ip}^W + A_T^L(W_i) v_{ip}^L + A_T(W_i) \psi_{ip} = 0 \quad \forall p \in \mathcal{P} \quad (37)$$

$$\sum_{i \in \mathcal{I}} A_D^W(W_i) v_{ip}^W + A_D(W_i) \psi_{ip} \leq \text{Max}_p^D \quad \forall p \in \mathcal{P} \quad (38)$$

$$\sum_{s \in \{bs, bs+1\}} \sum_{i \in \mathcal{I}} A_{Bs}^W(W_i) v_{ip}^W + A_{Bs}^L(W_i) v_{ip}^L + A_{Bs}(W_i) \psi_{ip} = 2D_d^{Bs} v_{bsp}^{Bs} \quad \forall bs \in \mathcal{BS}, p \in \mathcal{P} \quad (39)$$

$$W_{f\alpha}^S + \sum_{b \in \mathcal{B}} G_{bf}^\alpha w_{bp} + \sum_{t \in \mathcal{T}} G_{tf}^\alpha x_{tp} - \sum_{bs \in \mathcal{BS}} G_{bsf}^\alpha v_{bsp}^{Bs} = v_{f\alpha p}^S \quad \forall f \in \mathcal{F}, \alpha \in \{A, F\}, p \in \mathcal{P} \quad (40)$$

$$W_{f\alpha}^B + \sum_{b \in \mathcal{B}} A_{bf}^\alpha G_{bf}^\alpha w_{bp} + \sum_{t \in \mathcal{T}} A_{tf}^\alpha G_{tf}^\alpha x_{tp} - \sum_{bs \in \mathcal{BS}} A_{bsf}^\alpha G_{bsf}^\alpha v_{bsp}^B = v_{f\alpha p}^B \quad \forall f \in \mathcal{F}, \alpha \in \{A, F\}, p \in \mathcal{P} \quad (41)$$

$$\text{Min}_f^S \leq G_f v_{f,Fore,p}^S + (1 - G_f) v_{f,Aft,p}^S \leq \text{Max}_f^S \quad \forall f \in \mathcal{F}, p \in \mathcal{P} \quad (42)$$

$$\text{Min}_f^B \leq G_f v_{f,Fore,p}^B + (1 - G_f) v_{f,Aft,p}^B \leq \text{Max}_f^B \quad \forall f \in \mathcal{F}, p \in \mathcal{P} \quad (43)$$

Constraint (24) calculates the displacement of the vessel for every port. The next constraint, (25), sets the displacement interval variables together with (26) which ensures that exactly one displacement interval is active at each port. Constraint (27) and (28) defines v_{ip}^W to be equal to the displacement for the active displacement interval, and 0 for the rest. In a similar fashion (29) and (30) calculates the longitudinal centre of gravity (LCG). The centre of gravity is calculated as the sum of moments divided by the total displacement. The left-hand side of (30) calculates the sum of moments. This is done by considering the longitudinal centre of gravity for the tanks and blocks and multiplying with the weight stowed in these. The right-hand side uses the average weight of the displacement interval, instead of the actual displacement. Similar with v_{ip}^W v_{ip}^L is zero for the displacement intervals for which $\psi_i = 0$ and for the active displacement interval it lies within the bounds defined by constraint (29). Constraint (31)-(33) approximates the vertical centre of gravity (VCG). Constraint (31) and (32) defines bound for the vertical moment for each displacement interval, and in the case when $\psi_i = 1$ the two inequalities turn into an equality $v_p^V W_i = v_p^{VM}$. For the non-active displacement intervals these two constraints have no effect. Constraint (33)

calculates the vertical moment similar to how the LCG is calculated, but without multiplying the right-hand side with the weight. (34) ensures that the transversal centre of gravity is 0, meaning the middle of the vessel. Due to the construction of the blocks, the transversal centre of gravity is 0 for all blocks, and thus only the tanks and the moment of the empty vessel are considered in the calculation. Constraint (35)-(39) calculates the metacentre, trim, draft and buoyancy force using the linearization of the non-linear functions. Each of the planes for the functions is described using three factors, $A^W(W_i)$, $A^L(W_i)$ and $A(W_i)$. $A^W(W_i)$ is the displacement factor, $A^L(W_i)$ is the LCG factor, and $A(W_i)$ is the constant factor. (35) calculates the metacentre, and (36) defines the metacentric height to be greater than the minimum metacentric height allowed. In (37) the trim is required to be zero, and (38) enforces the draft be less than or equal to Max_p^D . As the trim is required to be zero, the draft does not depend on the LCG, but only the displacement of the vessel. Max_p^D is the minimum draft allowed when leaving port p , and will thus be the minimum of the draft at port p and port $p+1$. Constraint (39) calculates the buoyancy force (*bonjean*) between station bs and $bs+1$. The last four constraints (40)-(43) are related to the stress forces. The first two calculates the shearing and bending, and (42)-(43) defines the upper and lower bounds. The shear force on a vessel, at a given frame, is the integral of forces on either side of the frame, and the bending moment is the integral of moments on either side of the frame. The buoyancy forces are only approximated, and thus there is an accumulation of error when calculating the shear force and the bending moment. To reduce the impact of this error, constraint (40) and (41) respectively calculates the shear forces and bending moment with respect to the resulting forces acting fore and aft of the frame. Hence there are two shear variables for every frame at each port. Constraint (42) and (43) respectively sets the limits for the shear force and bending moment at each frame. The shear force and bending moment at a frame is estimated as a proportional calculation based on the position of the frame. This reduces the impact of the error accumulation as the fore-based computation is accurate in the bow and the aft-based computation is accurate in the stern. All these constraints ensure that the vessel is stable and can be declared seaworthy if the stacking rules are obeyed.

The set of constraints (24)-(43) defines a polyhedron with the feasible weight allocations. This polyhedron is denoted by \mathcal{W} in the model in Section 4.