IT University of Copenhagen

Ph.D. Thesis

Models and Algorithms for Container Vessel Stowage Optimization

Author:
Alberto Delgado

Supervisor:
Assoc.Prof. Rune Møller Jensen

December 22, 2013

To my wonderful wife, Annemarie

Acknowledgments

The writing of this dissertation and the culmination of my graduate studies have been possible thanks to the help, guidance, encouragement, and support of several wonderful people. First, I would like to thank my supervisor Rune Møller Jensen for giving me the opportunity to pursue a Ph.D. under his supervision. His constant support and constructive criticizing made the time I spent doing the Ph.D. and writing this thesis a very enriching experience. He also introduced me to the maritime shipping domain and showed me the importance of doing applied research, for which I am grateful. I specially thank Dario Pacino and Kevin Tierney, my friends and colleagues at the Decision Optimization Lab, with whom I had several insightful discussions during the past few years. Though I shared my office only with Kevin and Dario, they were not my only companions on this journey. I would also like to thank Peter Sestoft and all the other professors at the former SDG (Software Development Group) for insightful discussions on how to do research and the different ways to present it, and to all the Ph.D. students at SDG that like me felt the frustrations of missed deadlines, rejected papers, and bad reviews, but that also felt the satisfaction that doing research brings. Special thanks to Kosta, Helge, Mohammed, Juan, Joel, Josu, Paolo, and Hugo.

I also owe a big part of the successful culmination of this journey to my family. Annemarie, my wonderful wife, the one that with her love, patience and support filled with confidence and determination those moments full of doubts. My parents, my sister, and relatives that have always support me in the pursuing of my dreams, even when those would put an ocean between us, and to my family in law, that have helped me make of Denmark my new home. Special thanks also go to my colleagues from the Avispa research team in Colombia, where I started tasting the bitter sweetness of doing research, to Ange Optimization and the Ministry of Science, Innovation and Higher Education for partially funding my Ph.D., Maersk Line for providing us with valuable data and the domain knowledge needed to pursue our research goals, and to the IT University for all the support that provided me to finish my Ph.D.

Abstract

Containerized seaborne trade has played a key role in the transformation of the global economy in the last 50 years. In liner shipping companies, at the heart of this operation, several planning decisions are made based on the stowage capabilities of container vessels, from strategic decisions (e.g., selection of vessels to buy that satisfy specific demands), through to operational decisions (e.g., selection of containers that optimize revenue, and stowing those containers into a vessel). This thesis addresses the question of whether it is possible to formulate stowage optimization models, with acceptable trade-offs between accuracy and scalability, that can be used to assist solving planning problems in liner shipping companies. We consider two planning contexts to answer this question: Stowage Planning (SP) and the Cargo Composition Problem (CCP).

SP is the process of deciding where each container of those to be loaded in a port should be placed in a vessel, i.e., to generate *stowage plans*. This thesis explores two different approaches to solve this problem, both follow a 2-phase decomposition that assigns containers to vessel sections in the first phase, i.e., master planning, and that stows the containers assigned by the master planning to each vessel section and into vessel slots in the second phase, i.e., slot planning. The first SP approach automatically generates representative stowage plans. This dissertation presents an accurate mathematical programming model for master planning that includes features of stowage planning that have not been considered in previous work. For slot planning, a fast and accurate representative model to optimally stow vessel sections is introduced.
The second SP approach serves as the optimization component of a commercial decision support tool used for interactive planning of container vessels. Expert's know-how formulated as user preferences is integrated into the heuristic optimization component and used to tackle complex constraints and optimize combinatorial objectives. According to our experimental evaluation, stowage plans computed by our heuristic are competitive enough with respect to those made by experts under the same conditions.

The CCP evaluates how the stowage characteristics of containers with different features affect important performance measures used in liner shipping companies, e.g., vessel intake and cargo revenue. This dissertation provides the first problem description and introduces the first mathematical model with variable displacement. Specifically, a model that assigns containers to vessel slots, and which is actually solvable on real instances. To increase scalability, a master planning model with variable displacement to solve the CCP is also presented. Analysis on vessel intake and revenue optimization are performed on single and multi-port scenarios for a benchmark suite of ten vessels facilitated by our industrial collaborators.

Sets and Constants

Sets and constants common to more than one model

Sets	
B	Bonjean stations, excluding the station closest to stern.
F	Frames.
$\mathcal{L}$	Locations.
P	Scheduled ports to be visited by a vessel.
$\mathcal{T}$	Container types.
TR	Transports, where each transport $\langle p_i, p_j \rangle \in P^2$ is represented by a load-discharge port pair.
TR_p^{ON}	Transports $\langle p_i, p_j \rangle$ with containers on the vessel at departure ($p_i \leq p < p_j$) from port p.
U	Ballast tanks.
Constants	
$A^{\alpha}_{\{l,u,b\}f}$	Fore or aft distance from frame f to the longitudinal center of gravity of location l, ballast tank u, and buoyancy section between bonjean stations b and $b+1$.
D_l^L	Lcg of location l.
D_{pl}^V	Vcg of location l at port p.
D_b^B	Distance in meters between bonjean stations b and $b+1$ multiplied by water density.
$D_u^{\{T,V,L\}}$	Transversal (T), vertical (V), and longitudinal (L) center of gravity of ballast tank u.
K^{α}_{pl}	Capacity of *20*' and *40*' containers in location l at port p.
K^V_{pl}	Volume capacity in location l at port p.
K_{pl}^{TEU}	TEU capacity of location l at port p.
$K_{pl}^{\{RF,RC\}}$	Reefer plugs (RF) and reefer cells (RC) available in location l at port p.
K_u	Ballast water capacity of tank u.
LM_p^O	Longitudinal moment of the empty vessel including constant weights and release at port p.
LD_t^{τ}	Load-discharge matrix: number of containers of type τ on transport t.
$Min^{\{GM\}}$	Lower bound for metacentric height (GM).
$Min_f^{\{S,B\}}$	Lower bound for shear force (S) and bending moment (B) at frame f.
$Max_f^{\{S,B\}}$	Upper bound for shear force (S) and bending moment (B) at frame f.
$G^{\alpha}_{\{l,u,b\}f}$	Fraction of location l, ballast tank u, and buoyancy section between bonjean stations b and $b+1$ that lies fore or aft of frame f.
G_f	Fore-based fraction of frame f, where $G_f \in [0;1]$. $G_f = 1$ when f is the first frame at bow, and $G_f = 0$ when f is the first frame at stern.
VM_p^O	Vertical moment of the empty vessel including constant weights and release

	at port p.
W_{pl}^{α}	Weight capacity of *20'* and *40'* containers in location l at port p.
W_{p}^{O}	Constant weight of the vessel including release at port p.
$W_{f}^{S,\alpha}$	Constant weights fore or aft of frame f.
$W_{f}^{B,\alpha}$	Bending components of the constant weights fore or aft of frame f.
W_{τ}	Weight in metric tons of type τ.
Π_{τ}^{TEU}	TEUs needed to stow a container of type τ, i.e., $\Pi_{\tau}^{TEU} \in \{1, 2\}$.
Π_{τ}^{FEU}	FEUs needed to stow a container of type τ, i.e., $\Pi_{\tau}^{FEU} \in \{0.5, 1\}$.
Φ^{τ}	Volume required for type τ.

Sets and constants of slot planning

Sets	
$Classes$	Stack classes.
$Class^{i}$	Stacks of class i.
M	Tuple set of release containers and their corresponding cells indices: $\{(j, k, i) \mid j \in J, k \in K_j, i \in Q^{L}\}$.
J	Stacks.
Q	Containers to stow.
$Q^{\{V,L,R\}}$	Virtual (V), release (L), and reefer(R) containers.
$Q^{40\{A,F\}}$	*Aft40* and *Fore40 40'* containers. Q^{40A} and Q^{40F} have the same cardinality.
$Q^{\{20R,40R\}}$	*20'* and *40'* reefer containers.
$Q^{\neg R}$	Non-reefer *20'* and *40'* containers excluding virtual containers.
$Q^{\{20,40\}}$	*20'* and *40'* containers.
S	Slots.
$S^{\{A,F\}}$	Aft and Fore slots.
S_{j}	Slots in stack j.
$S_{j}^{\{A,F\}}$	Aft and Fore slots in stack j.
$S^{\{R,\neg R\}}$	Reefer (R) and non-reefer ($\neg R$) slots.
$S^{\neg RC}$	Slots of location l in cells with no reefer plugs.
$S^{\{20,40\}}$	Slots of location l with *20'* and *40'* capacity.
K_{j}	Cells in stack j.
Constants	
A_{ip}	Indicates whether container i is unloaded at port p. It is 1 if i is unloaded in port p, 0 otherwise.
H_{i}^{C}	Height in meters of container i.
H_{j}^{S}	Height limit in meters of stack j.
$H^{\{DC,HC\}}$	Height in meters of a normal (DC) and a high-cube (HC) container.
L_{i}^{C}	Length of container i.
P_{i}^{C}	Discharge port of container i.
$Q^{P=p,W=w}$	Containers with discharge port p and weight w to be stowed in location l.
$Q^{\{NC,HC\}}$	Normal (NC) and high-cube (HC) containers to be stowed in location l.
R_{i}^{C}	Indicates whether container i is reefer.
R_{jk}	Number of reefer plugs in cell k of stack j.
W_{i}^{C}	Weight in kilograms of container i.
W_{j}^{S}	Weight limit of stack j in kilograms.
C^{v}	Penalization for a container overstowing containers below.
C^{p}	Penalization for each different discharge port stowed in a stack.
C^{u}	Penalization for stowing containers in a stack.

C^r	Penalization for stowing a non-reefer container in a reefer slot.

Sets and constants of master planning

Constants

$A^W_{\{M,T,Bs\}}$	Weight coefficient for the linearization of metacenter (M), trim (T), and bonjean at station s (Bs).
$A^L_{\{M,T,Bs\}}$	Lcg coefficient for the linearization of metacenter (M), trim (T), and bonjean at station s (Bs).
$A_{\{M,T,Bs\}}$	Constant term for the linearization of metacenter (M), trim (T), and bonjean at station s (Bs).
$A^W_{\{DA,DF\}}$	Weight coefficient for the linearization of aft draft (DA) and fore draft (DF).
$A^L_{\{DA,DF\}}$	Lcg coefficient for the linearization of aft draft (DA) and fore draft (DF).
$A_{\{DA,DF\}}$	Constant term for the linearization of aft draft (DA) and fore draft (DF).
E_u	Initial condition of tank u.
$Min^{\{T,DA\}}$	Lower bound for trim (T) and aft draft (DA).
$Max^{\{T,DA,DF\}}$	Upper bound for trim (T), aft draft (DA), and fore draft (DA).
W_p	Approximated constant displacement at port p.
$W_{t\tau}$	Weight in metric tons of type τ during transport t.
ϵ_u	Allowed ballast change for tank u.

Sets and constants of heuristic optimization

Sets

$\mathcal{L}^O$	Locations with odd slots.
$\mathcal{L}_v$	Locations in bay v.
$\mathcal{L}_t$	Locations specified in the planning scenario to stow containers on transport t.
TR^l_p	Transports leaving from port p with destinations that have not been specified in the planning scenario as valid for location l.
V	Vessel bays.

Constants

$A^{\neg R}_{pl}$	Average volume gained by stowing a single *20*' non-reefer container in a non-reefer odd slot at port p.
A^R_{pl}	Average volume gained by stowing a single *20*' container in a reefer odd slot at port p.
C^τ_l	Profit for stowing a container of type τ in location l.
C^{HC}	Penalization for going over high-cube killing limit.
C^{OD}	Penalization for not using odd slots when there are *20*' available.
K^{40V}_{pl}	Volume capacity for stowing *40*' in location l at port p.
$K^{\neg RO}_{pl}$	Non-reefer odd slots in l at port p.
K^{OR}_{pl}	Reefer odd slots in l at port p.
K^O_{pl}	Odd slots in location l at port p.
K^O_t	Odd slots in locations specified in the planning scenario to stow containers of transport t ($\mathcal{L}^t$).
K^{O20}_t	Odd slots in $\mathcal{L}^t$ that we know in advance cannot be used due to lack of *20*' containers, i.e., $max(0, K^O_t - \sum_{\tau \in \mathcal{T}^{20}} LD^\tau_t)$.
K^K_{pl}	High-cube killing limit at port p: high-cube containers that can be stowed in l

	such that its slot capacity can still be completely utilized.
K_{pl}^{40N}	Volume capacity for stowing *40'* in stacks with no odd slots of location l at port p.
K_{pl}^{τ}	Containers of type τ that can be stowed in location l at port p.
W_{pv}^{u}	Upper weight limit specified in the planning scenario for bay v at port p.

Sets and constants of cargo composition

Sets

$\mathcal{C}$	Cells, i.e., $\mathcal{C} = \{(i,j,l) \mid i \in T_{lj},\ j \in J_l,\ l \in \mathcal{L}\}$.
$\mathcal{C}_l$	Cells in location $l \in \mathcal{L}$.
$\mathcal{C}_{lj}$	Cells in stack j and location l.
$\mathcal{C}^{\{\neg R, \neg 20, \neg 40\}}$	Cells with no reefer slots ($\neg R$), no *20'* capacity ($\neg 20$), and no *40'* capacity ($\neg 40$).
$\mathcal{C}_{\alpha}^{\neg R}$	Cells with no aft or fore reefer slot capacity.
$\mathcal{C}_c^-$	Cells below cell $c \in \mathcal{C}$.
$\mathcal{E}$	Aft and Fore slots of a cell, i.e., $\mathcal{E} = \{A,\ F\}$.
$\mathcal{E}^{\tau}$	Slots in a cell where containers of type τ can be stowed, i.e., $\{e \mid (\tau \in \mathcal{T}^{20} \wedge e \in \mathcal{E}) \vee (\tau \in \mathcal{T}^{40} \wedge e = A)\}$.
I	Displacement intervals.
J_l	Stacks in location l.
$\mathcal{L}^{ON}$	Locations on deck.
$\mathcal{L}_l^U$	Locations below location l.
$\mathcal{L}_n$	Locations in bin $n \in N$.
N	Bins of adjacent bays.
$TR_p^{\{A,OV\}}$	Transports $\langle p_i, p_j \rangle$ with containers loading or discharging ($p_i = p$ or $p_j = p$) at port p (A), or going over ($p_i < p < p_j$) port p (OV).

Constants

$A_{\{M,T,Bs\}}^{W}(W_i)$	Weight coefficient of displacement interval i for the linearization of metacenter (M), trim (T), and bonjean at station s (Bs).
$A_{\{M,T,Bs\}}^{L}(W_i)$	Lcg coefficient of displacement interval i for the linearization of metacenter (M), trim (T), and bonjean at station s (Bs).
$A_{\{M,T,Bs\}}(W_i)$	Constant term of displacement interval i for the linearization of metacenter (M), trim (T), and bonjean at station s (Bs).
c^+/c^-	Cell on top/below cell $c \in \mathcal{C}$. Let $c = c^+/c^-$ iff c is the top/below cell of the stack.
C_p^{OP}	Overstowage probability penalization at port p.
C_t^{τ}	Profit for stowing containers of type $\tau \in \mathcal{T}$ and transport $t \in TR$.
$C_p^{\{O,C,U\}}$	Overstowage (O), Makespan (C), and ballast water (U) penalization at port $p \in P$.
$D_c^{\{V,T\}}$	Vertical (V), Transversal (T) center of gravity of cell $c \in \mathcal{C}$.
H^{τ}	Height in meters of type τ.
H_{lj}	Height limit in meters of stack j in location l.
K_{pl}^{HC}	High-cube capacity in location l at port p.
M_c^S	BigM value for stack overstowage calculation: twice the number of slots below cell c.
$M_l^{\{H,U\}}$	BigM value for hatch overstowage calculations: twice the number of slots in location l (H) and twice the number of slots in locations below l (U).
Max^V	Maximum vertical moment possible for a vessel.
Max_i^L/Min_i^L	Maximum and minimum longitudinal center of gravity at displacement interval i.

TM_p^O	Transversal moment of the empty vessel including constant weights and release at port p.
W_{li}^{40}	*40*' weight capacity of stack $i \in J_l$ in location $l \in \mathcal{L}$.
W_{lie}^{20}	*20*' weight capacity of section $e \in \mathcal{E}$ in stack $i \in J_l$ location $l \in \mathcal{L}$.
$W_i^{\{-,+\}}$	Lower $(-)$ and upper $(+)$ bound of displacement interval i.
W_i	Average weight of displacement interval i.

Contents

1 Introduction **15**
1.1 Thesis Question . . . 19
1.2 Approach . . . 20
1.3 Thesis Contributions . . . 22
1.4 Publications . . . 23
1.5 Document Outline . . . 23

2 Container Stowage Planning **25**
2.1 Containers . . . 25
2.2 Container Vessels . . . 27
2.2.1 Physical Structure of a Container Vessel . . . 28
2.2.2 Vessel Stability and Stress Limits . . . 30
2.3 Maritime Container Terminals . . . 34
2.4 Container Stowage . . . 35
2.5 Stowage Plans . . . 37

3 Slot Planning **41**
3.1 Introduction . . . 41
3.2 Problem Statement . . . 42
3.3 Literature Review . . . 44
3.4 The Integer Programming (IP) Model . . . 45
3.5 Global constraint modeling . . . 48
3.6 The Constraint Programming (CP) Model . . . 49
3.6.1 Symmetry-Breaking and Implied Constraints . . . 52
3.6.2 Branching Strategies . . . 53
3.6.3 Lower Bounds . . . 54
3.7 Experiments . . . 56
3.7.1 Restowing Location Stowage Plans . . . 57
3.7.2 Slot Planning Master Plans . . . 62
3.8 Conclusions . . . 63

4 Accurate Master Planning **65**
4.1 Introduction . . . 65
4.2 Literature Review . . . 66
4.3 Stability and Stress Model with Ballast Tanks . . . 67
4.3.1 A Mixed Integer Programming (MIP) Model . . . 69
4.4 Analysis of Model Accuracy . . . 73
4.5 Conclusion . . . 77

5 Heuristic Optimization for Interactive Stowage Planning **79**
5.1 Introduction . . . 79
5.2 Problem Definition . . . 81
5.3 Angelstow . . . 81
5.4 Literature Review . . . 84
5.5 Heuristic Optimization Approach to Stowage Planning . . . 85
5.5.1 LP Model for Master Planning . . . 85
5.5.2 Greedy Algorithm for Slot Planning . . . 89
5.6 Experiments . . . 91
5.7 Conclusion . . . 96

6 Cargo Composition Problem **97**
6.1 Introduction . . . 97
6.2 Problem Definition . . . 98
6.3 Literature Review . . . 99
6.4 Complete Model for the Cargo Composition Problem (CCP) . . . 100
6.5 Decomposed model for the CCP . . . 107
6.6 Experiments . . . 110
6.6.1 Single Port Experiments . . . 111
6.6.2 Multi-Port Experiments . . . 120
6.7 Conclusions . . . 123

7 Conclusions **125**
7.1 Outlook and Future Directions . . . 126

Bibliography **129**

Chapter 1

Introduction

Modern shipping techniques have become the heart of globalized trade. They have transformed business models around the world by connecting new suppliers and consumers thousands of kilometers away from each other. The mechanization of sea transport replacing expensive human labor with more efficient equipment, and the development of larger and more efficient ships to achieve economy of scales have been key factors in this transformation. Equally or more importantly, however, has been the introduction of *containerization*: the transportation of cargo in standard sized boxes known as *containers*. Containerization provided a standard interface between the different parties involved in the intermodal freight transport chain, allowing the development of an integrated transport system.

Seaborne trade accounts for over 80% of the global trade by volume and over 70% by value. Containerized trade has grown steadily, by almost 14 times its original volume between 1980 and 2011 (UNCTAD (2012)), reaching 15.8% of all cargo traded by volume and 52%[1] by value in 2011. Though the world economy is still in recovery, the growth in seaborne container trade is expected to continue, driven mostly by increasing demand for manufactured goods in emerging economies and by containerization of unconventional commodities, e.g., scrap steel and recycled paper (UNCTAD (2012)). The majority of seaborne containers are transported by *container vessels*, cellular vessels specifically designed to carry containers. These vessels are operated by *liner shipping companies* [2], with a fleet capacity that accounts for 97.2% of the total active capacity deployed on liner trades (Alphaliner (2013)).

Even with the magnitude of their operation, liner companies face a highly competitive market. The liner shipping sector, for instance, is far from being consolidated. By January 2012, almost 70% of the Twenty-foot Equivalent Unit (TEU)[3] capacity was in

[1] From the website of the World Shipping Council (WSC) at www.worldshipping.org. Accessed May 2013.

[2] Liner shipping companies, carrier, container carriers, and liner companies will be used interchangeably in this thesis.

[3] Measure used in the shipping industry for volume capacity. One TEU is equivalent to the dimensions of a standard *20*' container.

the hands of the top twenty shipping companies, with the three largest companies controlling almost 30% (UNCTAD (2012)). This considerable number of relevant players, together with the low segmentation of the market and other external factors (e.g., the financial crisis in 2009), has increased the need for carriers to be more competitive. To do so, different strategies have been devised. One strategy has been to reduce transportation costs per TEU by increasing vessel capacity (i.e., economies of scale), with vessels reaching sizes of up to 18,270 TEU[4]. This approach, however, is limited by the demand on the trade routes, since the vessels must be filled to a certain capacity to achieve savings by economies of scale. Furthermore, the financial crises of 2009 reduced the demand for freight transport, and with most of the carriers attempting to reduce costs by acquiring larger vessels, the market was flooded with capacity. By 2013, however, part of the extra capacity has been absorbed by the implementation of the *slow steaming* policy. This policy aims at reducing bunker consumption by having container vessels sailing at lower speeds. This speed reduction forces the addition of new vessels to the trade lanes in order to maintain the same service frequency. Though this policy has provided some relief to the over capacitated market it has not absorbed all the extra capacity available, keeping freight rates at a very low level.

In another strategy to be more competitive, the liner shipping companies tried to optimize the different planning problems they face. Traditionally, planning problems in the liner shipping industry can be classified as *strategic*, *tactical*, or *operational*, depending on their planning horizon (Christiansen et al. (2007)). Figure 1.1 depicts the three planning levels and some of the planning problems associated with each of them. At the *strategic level*, decisions involving major economical investments and commitments are made. These decisions cover a medium to long-term horizon that oscillates between one to five years, but which can be extended by five to ten years, depending on the decision (designing/buying a vessel, building a container terminal). The decisions at this level shape the tactical and operational decisions to be made in the future. Four of the most important *strategic decisions* are: *selecting trade lanes* the carrier wishes to operate, *selecting target market share* for each trade lane defining the size of the operation the carrier wants, *choosing the fleet mix and size* necessary to satisfy the demand of the trade lanes and market share chosen by the carrier, and *choosing main ports* and transshipment hubs to call in the trade lanes previously chosen (Kjeldsen (2012)).

Tactical decisions cover a medium term horizon, spanning from two to three months to up to a year. There are three important decisions made at the tactical level: *route design*, where the carrier decides sequences of ports to call in order to cover the trade lanes chosen at the strategic level; *vessel deployment*, where vessels are assigned to each route in order to satisfy the forecasted demand; and *scheduling*, where the carrier assigns a time to call each port to the vessels deployed in the different routes (Andersen (2010)).

[4]Capacity of the Triple-E class, to be delivered in June 2013

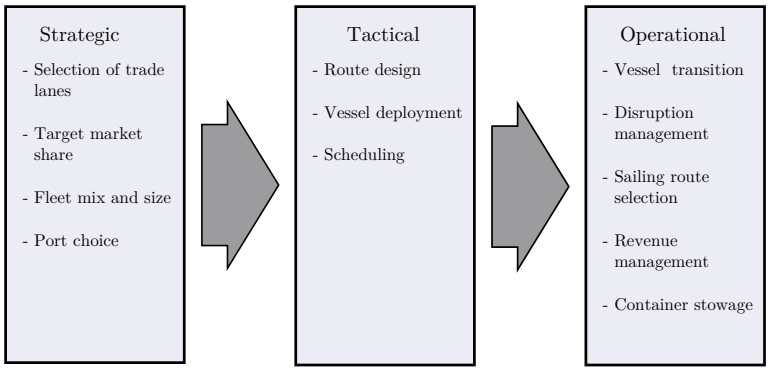


Figure 1.1: Planning levels and their activities in liner shipping companies. The arrows indicate the direction of influence of each level.

Finally, *operational decisions* cover a short term horizon, spanning from a few hours to a few months. Decisions at this level either relate to container vessels or to cargo. The most common operational decisions related to vessels are: *vessel transitions*, such as deciding when and how vessels will move to a new schedule from an old one; *disruption management*, where the vessel route or its sailing speed is modified in order to cope with delays in the schedule; and *sailing route selection*, where the sailing route and speed is decided based on weather factors and sea currents in order to reduce the cost of the trip, (Christiansen et al. (2007)). With respect to operational decisions related to cargo, the most common decisions are *revenue management* and *container stowage*. In revenue management, *route managers* decide what kind of cargo is more profitable to transport on a route based on the available capacity of the vessels currently sailing, yield analysis, and forecast demand. They provide this information to the sales department, who will take cargo based on this analysis. Then, *capacity managers* decide which containers to load onto the vessel from those taken by the sales department. This is based on the revenue they generate and their shipping priority. In container stowage, on the other hand, *stowage coordinators* decide where to stow the containers that the capacity manager has decided to take onto the container vessel.

Among many of the decisions made when solving planning problems in a liner shipping company, a common denominator is the need to consider container vessels and their TEU capacity. For instance, at the strategic level, the carrier purchases the vessels with the capacity and other features needed to satisfy the demand for the trade lanes they plan to operate. At a tactical level, vessels are deployed onto routes based on their capacity and other features in order to satisfy the demand of each route. At the operational level, route managers segment the cargo based on its yield, expected demand, and on how much of the demand the vessels will be able to transport. Moreover, capacity managers accept and reject containers based on the current capacity of

the container vessels.

Most of these decisions are usually made under the assumption that a container vessel can be filled up with containers of any kind, and that they can be freely stowed in the vessel. This assumption, however, is far from reality. Stowing containers in a vessel is a complex problem. High level constraints force containers to be assigned to vessel sections in such a way that stability and stress forces are kept within limits, while low level stacking rules and constraints force container stacks to follow valid patterns and to be within capacity limits. Thus, in a number of planning problems in the shipping industry, having a clear picture of the viability of stowing cargo compositions in a container vessel is crucial. Inaccuracies at this level lead to a sub-optimal utilization of the fleet by over/under-estimating its capabilities, keeping carriers from achieving economies of scale, and missing opportunities in the market by not satisfying demands. It is highly relevant for shipping companies to have at hand accurate stowage models that provide them with stowage footprints of container vessels, allowing them to perform different kinds of optimizations. Figure 1.2 shows some of the planning problems where accurate stowage modelling and optimization can be used.

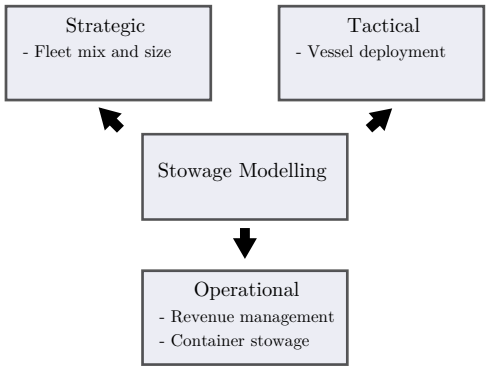


Figure 1.2: Planning decisions where stowage modelling can be used.

Previous work

The academic community has been interested in stowage modelling since the 1970s. Their work, however, has mainly focused on *stowage planning*, i.e., the process of generating *stowage plans* that assign containers to vessel slots. Academic work can be divided into two main categories: single-phase and multi-phase approaches. Single-phase approaches have a plain representation of the stowage planning problem, (e.g., stow containers into vessel slots). A common denominator among these approaches is to sacrifice model accuracy to achieve scalability. For instance, Botter and Brinati (1992), introduced an IP model of the first accurate but intractable formulation of the stowage planning problem, and more recently, Li et al. (2008), introduced an IP model to solve a yet more simplified version of the stowage planning problem. Multi-phase

approaches decompose the problem hierarchically into two or more layers of abstraction (e.g., first assign containers to vessel sub-sections, then stow them into vessel slots), and they are currently the most successful in terms of model accuracy and scalability. Wilson and Roach (1999), introduced a model that includes several major aspects of stowage planning in a 2-phase approach, resembling the workflow of stowage coordinators. Other approaches inspired by similar decompositions include the recent work of Ambrosino et al. (2010), and Pacino et al. (2011).

Though on a smaller scale than in stowage planning, there is also academic work attempting to use stowage modelling and optimization to make planning decisions on other levels of the shipping companies. Most of the work has focused on revenue management (e.g., Zurheide and Fischer (2012); Feng and Chang (2008); Ting and Tzeng (2004)), with models that select which containers to load on container vessels deployed on a route, based on the revenue they generate. These models, however, consider very basic volume and weight capacity constraints for containers with a few different features, disregarding all other aspects of stowage.

Academic work on stowage modelling and its application for liner shipping is considerable; its impact, however, has been limited. The stowage of container vessels is a problem full of corner cases and subtle details. The lack of a strong industrial collaboration has made it difficult for academics to access detailed information on the vessel designs of key importance for the formulation of accurate models. It has also been difficult to define representative problems that include the most relevant features of a stowage model identified by the industry. This has led to most of the published approaches to have a trade-off between accuracy and scalability that is not practical for the industrial setup of a stowage model. Either execution times running too high or problem abstractions miss several key features of the problem.

1.1 Thesis Question

From a long term collaboration with a liner shipping company we have devised a set of key features that define the level of accuracy and scalability stowage models must have to be used in an industrial setup. Nonetheless, even when only considering the key features defined with our industrial collaborator, solving stowage models in a scalable manner remains a challenging problem. Moreover, as mentioned before, most of the research on stowage modelling focuses on stowage planning, with almost no academic work on using accurate stowage modelling for other planning decisions in liner shipping companies. The few of these decisions for which optimization models have been investigated, e.g., revenue management, consider over simplified stowage constraints that reduce their applicability. Thus, the challenges in stowage modelling that this thesis addresses can be summarized in the question below:

> *Can stowage optimization models, with an acceptable trade-off between accuracy and scalability, be formulated to assist solving different planning prob-*

lems in liner shipping companies?

1.2 Approach

This thesis addresses the above research question by formulating stowage optimization models that support two different planning contexts within a liner shipping company: stowage planning and the cargo composition problem. The acceptable trade-off between accuracy and scalability is defined by our industrial collaborators and depends on the industrial setup where the stowage models are meant to be deployed within the company.

Stowage planning is the most straightforward application of stowage modelling in liner shipping companies. This thesis investigates two scalable stowage approaches to generate stowage plans that follow a 2-phase hierarchical decomposition, similar to that of the most successful approaches (Wilson and Roach (2000); Kang and Kim (2002); Ambrosino et al. (2010); Guilbert and Paquin (2010)), resembling the workflow of stowage coordinators (see Figure 1.3). The first phase of this decomposition, *master planning*, assigns containers to load in the port to vessel sections. It considers high-level constraints and objectives such as stability, vessel sections' capacities, and the homogeneous distribution of crane workload along the vessel. The second phase, *slot planning*, follows the assignment made in the master planning phase and assigns the containers to load each vessel section to vessel slots. It considers low level stacking constraints and rules of thumb to generate robust stowage plans.

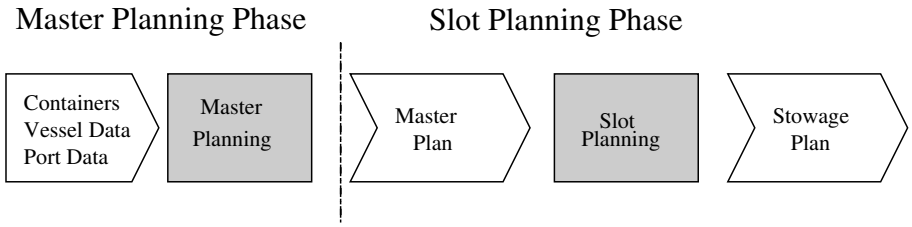


Figure 1.3: 2-phase hierarchical decomposition of stowage planning.

Both of the two stowage approaches studied in this thesis have different trade-offs between accuracy and scalability, based on the industrial setup where they are to be deployed. The first approach automatically generates stowage plans and includes all key aspects of stowage planning defined by our industrial collaborators. Since the approach solves a representative problem, the stowage plans that it generates cannot be used directly in practice. Stowage coordinators use these plans as initial solutions that require few modifications to turn into final stowage plans. With respect to scalability, according to our collaborator these plans must be generated within a time limit of 10 minutes. The reason being that stowage coordinators have only a few hours to generate stowage plans and within this short time, they must be able to try out several different forecast scenarios.

During master planning, containers are assigned to vessel sections by a MIP model (Pacino, Delgado, Jensen, and Bebbington (2012), Delgado et al. (2013)), on the proviso that the vessel is seaworthy, i.e., stability and stress forces are within limits, high level objectives to reduce time at port are optimized, and capacity constraints in each vessel section are fulfilled. Given that we are solving the master planning phase with a MIP model, a fractional number of containers may be assigned to vessel sections. Though it is not possible to divide containers, we show, experimentally, that the impact of this is negligible. During the slot planning phase, the CP model and Local Search (LS) algorithm presented in Delgado et al. (2012) and Pacino and Jensen (2012), respectively, stow containers into vessel slots in each vessel section, following the assignment made by the master planning phase. Containers are stowed into container stacks that follow specific stacking rules and satisfy capacity constraints, while optimizing objectives aiming at producing robust stowage plans.

The second approach for stowage planning is the optimization component of a commercial decision support tool to stow container vessels called *Angelstow*[5], (Delgado et al. (2012)). This approach demands a different trade-off between accuracy and scalability due to the interactive nature of the decision support tool. Accuracy is reduced in order to provide fast stowage plans, since the time limit defined for the generation of the plans is two seconds, on average. The master planning phase of this approach integrates expert know-how provided by the stowage coordinator into a Linear Programmming (LP) model that considers a reduced set of key aspects of stowage planning. The know-how is encapsulated into a *planning scenario*, where the stowage coordinator provides experience on how to address stability and stress forces, and reduce time at port. The slot planning phase is solved by a greedy algorithm that builds container stacks, satisfying stacking constraints and rules.

The second planning context that we consider in this thesis is the cargo composition problem. Here we evaluate how the stowage characteristics of containers with different features affect important performance measures used in liner shipping companies, e.g., vessel intake and cargo revenue. We introduced the first complete model that stows containers into vessel slots and solves real instances. To improve scalability, we present a model based on the master planning phase of the 2-phase decomposition depicted in Figure 1.3, to generate fast sub-optimal solutions.

Stowage planning and the cargo composition problem consider the same constraints with respect to stowing containers and vessel seaworthiness. There is, however, a fundamental difference. In stowage planning, the vessel displacement is known in advance and only small changes are expected to happen, since the loadlist of containers has been already decided. This allows an efficient implementation of accurate stability and stress forces calculations. In the case of the cargo composition problem, however, part of the problem is to determine the containers to be stowed, which makes the vessel dis-

[5]Ange Optimization ApS, available for purchase at www.http://ange.dk/main/angelstow.

placement variable. We address this issue by providing the proposed models for solving the cargo composition problem, with the capability to efficiently calculate stability and stress forces in a variable displacement environment, and at a small performance cost.

1.3 Thesis Contributions

To address the research question posed above, this thesis presents the following five major contributions:

1. **Optimal slot planning**
   This thesis presents an accurate representative model of slot planning of vessel sections, and introduces (to our knowledge), the fastest approach to generate optimal slot plans. The model successfully finds optimal solutions for most of the industrial instances where it has been tested and was successfully deployed in a planning module for the automatic generation of stowage plans.

2. **Optimization module for interactive stowage planning**
   This thesis introduces the first heuristic optimization module that incorporates expert's know-how into the fast generation of stowage plans. Moreover, the module has been successfully integrated into a commercial decision support used for stowage planning tools and its accuracy tested with industrial data.

3. **Accurate scalable master planning**
   This thesis introduces an accurate optimization model for master planning. It includes key aspects of stability and stress forces, and the modelling of ballast tanks, (i.e., tanks loading ballast water used to adjust vessel stability). The model is tested on industrial data and scales to larger instances than previously tested in the literature.

4. **Variable displacement model for master planning**
   This thesis introduces the first master planning model with accurate stability and stress calculations that efficiently considers variable displacement of container vessels.

5. **Cargo composition problem**
   This thesis introduces a new problem into the liner shipping domain and the first model with variable displacement that assigns containers to slots, that is actually solvable on real instances. This model is also the first attempt to use accurate stowage modelling for decision problems other than stowage planning within a liner shipping company.

1.4 Publications

1. Delgado, Alberto; Møller Jensen, Rune; Janstrup, Kira; Høyer Rose,Trine and Høj Andersen, Kent. **A constraint programming model for fast optimal stowage of container vessel bays**. European Journal of Operational Research 220, no. 1 (2012): 251-261.

2. Delgado, Alberto; Møller Jensen, Rune and Schulte, Christian. **Generating optimal stowage plans for container vessel bays**. Principles and Practice of Constraint Programming-CP 2009, pp. 6-20. Springer Berlin Heidelberg, 2009.

3. Delgado, Alberto; Møller Jensen, Rune and Guilbert, Nicolas. **A placement heuristic for a commercial decision support system for container vessel stowage**. Conferencia Latinoamericana en Informatica (CLEI), 2012 XXXVIII, pp. 1-9. IEEE, 2012.

4. Dario, Pacino; Delgado, Alberto; Møller Jensen, Rune and Bebbington, Tom. **An accurate model for seaworthy container vessel stowage planning with ballast tanks**. Computational Logistics, pp. 17-32. Springer Berlin Heidelberg, 2012.

5. Delgado, Alberto; Pacino, Dario and Møller Jensen, Rune. **Container vessel stowage planning with ballast tanks**. Extended abstract to appear in proceedings of Second INFORMS Transportation Science and Logistics Society Workshop, (2013 TSL Workshop)

6. Delgado, Alberto, and Rune Møller Jensen. **Cargo composition analysis in container vessels**. To be submitted to the European Journal of Operational Research

7. Delgado, Alberto; Møller Jensen, Rune and Guilbert, Nicolas. **AngelStow: A commercial optimization-based decision support tool for stowage planning**. Presentation at 2nd International Conference on Computational Logistics (ICCL11), 2011.

8. Pacino, Dario; Delgado, Alberto; Møller Jensen, Rune and Bebbington, Tom. **Fast generation of near-optimal plans for eco-efficient stowage of large container vessels**. In Computational Logistics, pp. 286-301. Springer Berlin Heidelberg, 2011.

1.5 Document Outline

Chapter 2 Container Stowage Planning. In this chapter we introduce the reader to the liner shipping domain, providing the terminology and knowledge necessary

to understand the contributions of the thesis. The chapter is based on the knowledge gathered from our industrial collaborators, work experience in the domain, and the book on ship stability by Barrass and Derrett (2011).

Chapter 3 Slot Planning. In this chapter we introduce a representative problem for the slot planning of vessel sections and two optimization models for solving this problem: a CP and an IP model. We present a detailed description of both models and experimental comparisons between them using industrial instances.
The chapter is based on the papers by Delgado et al. (2009) and Delgado et al. (2012). Note that the defendant is the main contributor of the papers and was in charge of the implementation of the CP model.

Chapter 4 Accurate Master Planning. In this chapter we introduce a MIP model for master planning that includes accurate stability and stress calculations, and ballast tanks. We propose an approach to tackle the non-linearities introduced by the variable displacement due to the tanks, and experimentally evaluate the impact of this approach in the vessel stability and stress forces calculations.
The chapter is based on the paper by Pacino, Delgado, Jensen, and Bebbington (2012). Note that the defendant made significant contributions to the formulation of the model, experimental section of the paper, and the implementation of the models.

Chapter 5 Heuristic Optimization for Interactive Stowage Planning. This chapter presents the optimization component of a decision support tool used for the interactive generation of stowage plans. The heuristic we introduce here generates stowage plans within seconds following the recommendations from an expert user, (stowage coordinator).
The chapter is based on the paper by Delgado et al. (2012). Note that the defendant is the main contributor to the paper and is responsible for the implementation of the methods.

Chapter 6 Cargo Composition Problem. This chapter presents a model that assigns containers to vessel slots, and an extension to the model presented in chapter 4, to generate cargo compositions, optimize vessel intake, and cargo revenue. Both models allow accurate stability and stress calculations when the vessel displacement is variable.
The chapter is based on the paper Delgado and Jensen (2013). Note that the defendant is the main contributor to the paper and is responsible for the implementation of the models.

Chapter 7 Conclusions. This chapter presents the conclusions of the thesis and future research directions in stowage modelling and, to wit, its applications in the liner shipping domain.

Chapter 2

Container Stowage Planning

Commercial shipping can be divided into three segments according to the type and volume of the transported commodities: *bulk shipping*, *liner shipping*, and *specialized shipping* (Stopford (2009)). *Bulk shipping* transports bulk homogeneous cargo in big volumes, most of the times raw materials, (e.g., crude oil, iron and grain). *Liner shipping* transports different types of general cargo in small volumes. It includes containerized cargo, palletized cargo, and liquid cargo stored in small tanks, among others. *Specialized shipping* deals with bulk and general cargo in big volumes. Vessels are specially designed for the kind of cargo they transport. It includes the transportation of motor vehicles, refrigerated foods, and different types of liquid gases that demand specific tankers.

In the liner shipping domain, general cargo is loaded into standard size boxes called *containers*. Containers are mainly transported in *container vessels*. Container vessels are assigned to *services*, fixed routes of ports where the vessels call and where containers are loaded and unloaded, (similar to a bus service). A vessel visits each port in a service following a schedule and must sail even if it is only partially full. Liner shipping with container vessels is the domain assumed in this thesis.

2.1 Containers

A container is a standardized piece of equipment. Containers have been designed to facilitate their transfer among different transportation modes, e.g., container ship, train, truck, etc., without intermediate transfer of their content. They are built to support standard shipping, storing, and handling conditions so they can be used multiple times. Their lifespan depends on the conditions they have been exposed to and their usage level. It ranges from ten to fifteen years, in average.

Different types of containers are available to fulfill specific requirements of cargo. The most common containers are box-shape *20'* or *40'* long, and 8' wide. They are either 8'6" (*standard*) or 9'6" (*high-cube*) high, though it is uncommon to see *20'* high-cube containers. Standard *20'* and *40'* containers, together with *40'* high-cube con-

tainers, account for almost 90% of the world's container inventory in circulation, with *20'* standard and *40'* high-cube containers, depicted in Figure 2.1, being the 30% and 45.7% of the total TEU backlog in 2011, respectively, (Drewry Maritime Research (2011)). Containers of other lengths are also available, e.g., *10'*, *45'*, *48'*, *53'*, the most

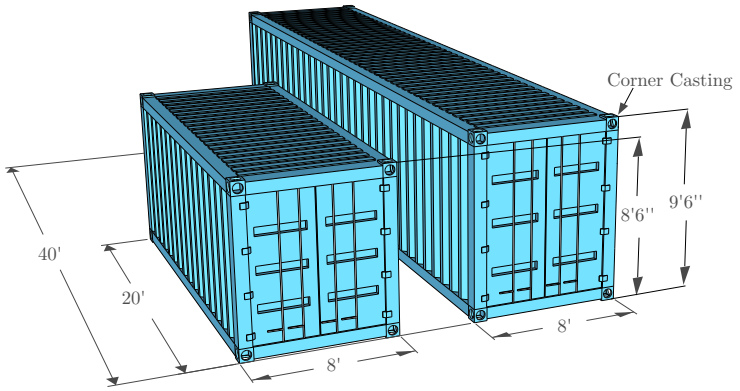


Figure 2.1: Dimensions of the most common containers in the fleet, adapted from Pacino (2012).

common among these lengths being the *45'* container. Most of the container vessels, however, are designed mainly to handle *20'*, *40'*, and to some extent *45'* containers. Nowadays, the kind of cargo that container vessels transport go beyond the capabilities of standard box-shaped containers, thus, containers for cargo with special requirements are needed. For instance, cargo that needs constant ventilation, e.g., organic products, use *ventilated* containers that are provided with small openings for air circulation. Cargo that is temperature sensitive and needs to be kept above or below freezing point, e.g., perishable goods, is transported in *reefer* and *insulated* containers. Reefer containers are provided with electric cooling engines and need to be stored close to a power plug, while insulated containers need to be provided with air delivered to the correct temperature. Though ventilated containers have more capacity than reefer containers (no cooling engine), their transportation and storage is more demanding since they need an external unit that provides them with air. Dry bulk cargo (e.g., grain and seeds), is transported in *bulk* containers. These containers are provided with three manholes at the top to facilitate their loading and with discharge openings in each door wing. Separated *tank* container fleets are available for the transport of chemical products and foodstuff. For odd shaped cargo *Out-of-gauge* (OOG) containers are available. Among these containers we find the *hardtop* containers with removable roof, that can be loaded by crane and are specially designed for heavy and high loads; the *open top* containers, specially designed for overheight cargo; the *half height open top 20'* containers; the *flat* containers with only the front, back, and bottom walls that are designed for heavy loads and over-width cargo; as well as the *platform* containers with no walls which are designed specifically for heavy loads an over-sized cargo. Containers carrying hazardous

goods are divided into four *IMO* risk groups based on their content. Each group has a set of rules specified in regulations (e.g., Hamburg (2008)), that describe how they can be stowed, i.e., safety distances to other IMO containers, and in which sections of the container vessel they may be stowed.

2.2 Container Vessels

Container vessels are cellular ships designed and constructed with the sole purpose of transporting containers. Since the appearance of the first container vessel, the *Ideal-X*[1] in 1956, container vessel capacity has been continuously increasing to reach an astonishing 18,270 TEU in 2013, (*Triple-E Class*). This growth has been driven by the principle of economy of scale in the shipping industry, the increasing demand of goods, and the ever growing number of commodities that can be transported in containers. Figure 2.2 shows the increase in TEU capacity of container vessels in the last fifteen years. By 2013, the total active capacity deployed on liner trades around the world was

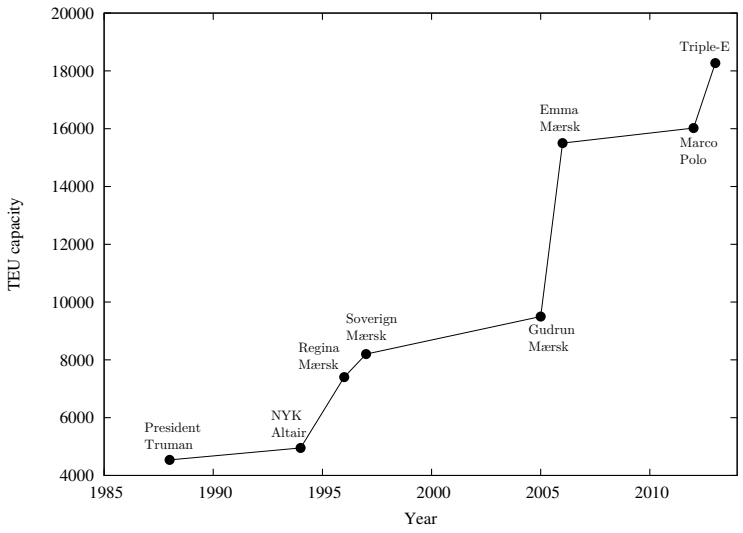


Figure 2.2: Growing TEU capacity of container vessels from 1985 to 2013

16.88 million of TEU carried by 5,929 ships, with 4,951 container vessels accounting for 97.2% of this capacity. Container vessels can be classified according to their TEU capacity and hull dimensions in under the following categories (MAN Diesel & Turbo):

- **Small Feeder** with up to 1,000 TEU capacity and a beam of 23m. They are usually used for short-sea transportation

[1] A transformed tanker carrying fifty-eight *35*' containers.

- **Feeder** ranging from 1,001 to 2,800 TEU capacity and beam size between 23 to 30.2m. These vessels feed the bigger container vessels and serve areas with low container demand
- **Panamax** with capacity from 2,800 to 5,100 TEU, with a max. beam of 32.3m and a max. draught of 12m. They were the biggest container vessels produced until 1988, its size matches that of the Panama canal at that time
- **Post Panamax** with capacity from 5,500 to 10,000 TEU, with bean sizes from 39.8 to 45.6m. These vessels were the first to exceed the 32.3m beam size limit of the Panama canal in 1988
- **New Panamax** with capacity from 12,000 to 14500 TEU, max. beam of 48m, max. draught of 15.2m, and max. length of 365.8m. These are vessels constrained by the dimension of the new Panama canal, to be operational in 2015
- **ULCV** (Ultra Large Container Vessels), with capacity from 14,500 and with beams greater than 48.8m. These vessels are larger than the new Panama Canal, to go operational in 2015.

Nowadays the world fleet consists of 39.18% of feeder vessels, 20.76% of Panamax, 20.34% of small vessels, 16.42% of Post Panamax, and 3.3% of ULCV. TEU-wise, however, the distribution is different. Panamax vessels account for the 26.46% of the fleet TEU capacity, Post Panamax vessels for the 35.5%, Feeder vessels for the 21.45%, ULCV for the 12.7%, and Small Feeder vessels for the remaining 3.94%.

2.2.1 Physical Structure of a Container Vessel

The area of a container vessel where containers are stowed is longitudinally divided into sections called *bays*, (See Figure 2.3). Bays are divided in two sections, on deck and

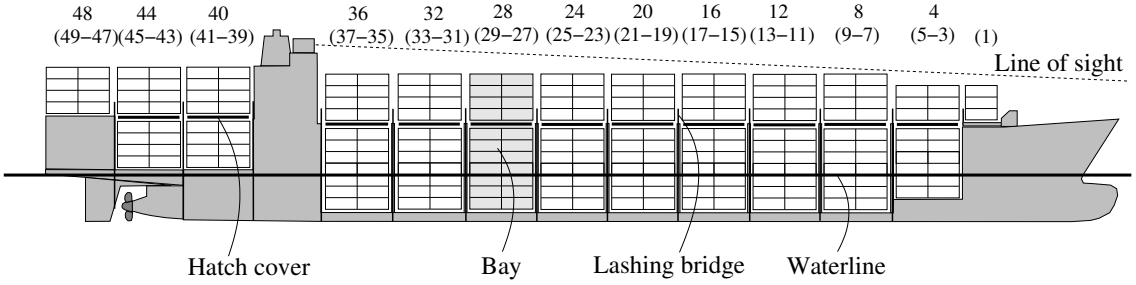
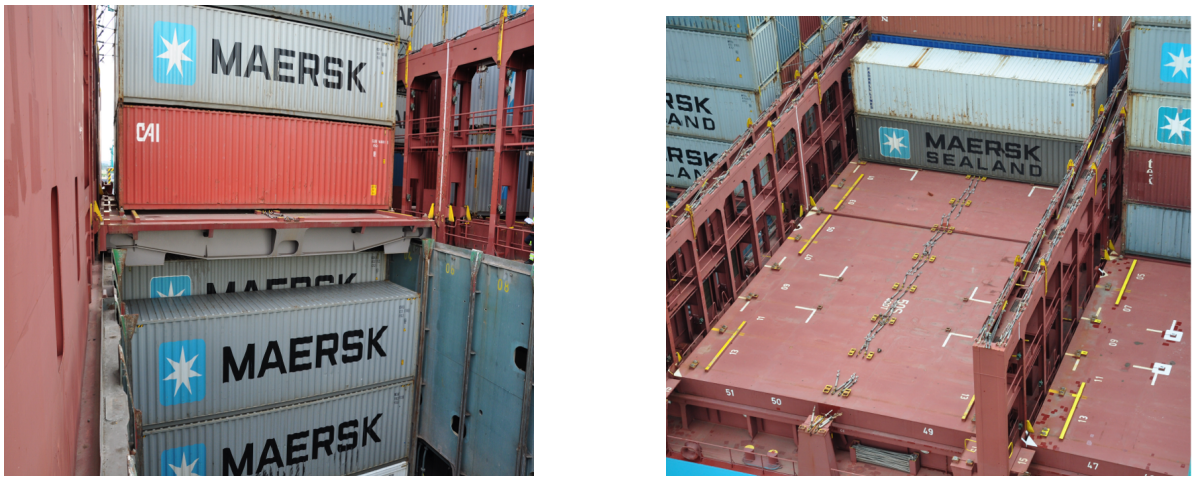


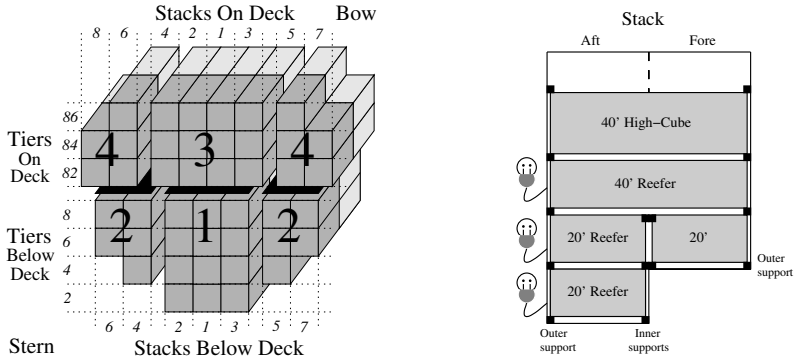
Figure 2.3: The arrangement of cargo space in a container vessel.

below deck, by a number of flat leak-proof structures called *hatch covers*. Figure 2.4 (a) shows a hatch cover dividing a bay section. Containers are stowed on top of and below the hatch cover. The below deck section of a bay is also known as *cargo hold*. Each sub-section of a bay consists of a row of stacks one container wide, (See Figure 2.5 (a)).



(a) (b)

Figure 2.4: (a) Lateral view of a bay. The hatch cover separates the on deck and below deck section. Containers are stowed in both sections. The picture also shows the cell guides below deck. (b) Two sections of the lashing bridge with hatch covers in between. The picture also shows containers on top of the hatch covers, secured to the lashing bridge.



(a) (b)

Figure 2.5: (a) A front view of a vessel bay. (b) A side view of a partially loaded stack. Each power plug represents a reefer slot. Reefer containers are drawn with electric cords.

Stacks either on or below deck are grouped into storage areas called *locations*. A *location* is a set of stacks, not necessarily consecutive, that coincide with the hatch covers. Pairs of storage areas on the starboard and the port side of the vessel are merged into single locations to ease stability calculation, (see locations 2 and 4 in Figure 2.5 (a)). Note that locations are not physically defined, but are abstractions used in practice to aid the process of stowing containers in the vessel. A stack holds vertically arranged *cells* indexed by *tiers*. Cells are divided into two *slots*, the Aft (closest to the stern), and Fore, (closest to the bow). A slot can hold a *20'* container and two slots in the same cell can hold a *40'* or a *45'* container. Due to the shape of the hull some cells in the vessel have only a single slot available, these cells are known as *odd cells*, (See Figure 2.5

(b)). Some slots have extra features that allow the stowing of containers with special demands, e.g, slots with electric plugs provide refrigerated containers with electricity for their engines. Figure 2.5 (b) depicts the layout of a stack.

Stacks have height and weight limits. Two weight limits exist for each stack, one regarding the outer container supports and one regarding the inner supports. Limits on the inner supports are often the smallest, as the vessel structure in the middle of a stack is weaker. The inner supports are used only when *20'* containers are stowed, as depicted in Figure 2.5 (b). When *20'* and *40'* containers are mixed in the same stack, only half of the *20'* weight is considered to be supported by the outer supports, since the other half sits on the inner supports. Below deck, cell guides secure containers transversely, (see Figure 2.4 (a)). Containers on deck are secured to each other, to the deck, or to the *lashing bridge* - a fixed steel structure running over deck across the ship that has been designed to provide container stacks with extra support, allowing them to stack higher. Figure 2.4 (b) depicts two sections of the lashing bridge separated by hatch covers.

Ballast tanks are carefully located tanks along the vessel that are used to modify displacement and stability conditions of the vessel by changing the amount of water they hold. Some of these tanks are situated on the sides of the vessel and act as heeling tanks when containers are being loaded and unloaded in port.

Containers in a vessel are identified by the bay, stack and tier of the cell where they are stowed. The industry uses a special indexing system that allows them to quickly narrow down the area where containers are stowed. Each physical bay is identified by three consecutive numerical indices, two odd and one even, (See Figure 2.3). The even index is used to identify the *40'* containers stowed in the bay, while the smallest odd index identifies *20'* containers stowed in the fore section of the bay, and the greatest odd index identifies *20'* containers in the aft section. As depicted in Figure 2.5 (b), tiers indices are even numbers that, on deck, start with the number 82, while below deck they start with number 2. Stack indices begin from the middle out, with the even indices towards starboard and the odd indices towards the port.

2.2.2 Vessel Stability and Stress Limits[2]

For a vessel to leave port it must be deemed seaworthy. All items loaded, i.e., cargo, ballast water, fuel, etc., must be distributed along the vessel such that stability values are acceptable and the stresses on the vessel structure are within limits. The *displacement* of a vessel (its total weight), determines the portion of the hull that is under water. The *water line* is the line where the water meets the hull of the vessel, and the plane that cuts the hull at the water line is called *water plane*. The *design waterline* of a vessel is the water line at the point where the vessel is loaded at its full capacity. The vertical line passing through the interception of the forward surface of the *stem* (most forward part of the vessel), with the design waterline is known as the *fore perpendicular*, while

[2]Inspired by Barrass and Derrett (2011)

the *aft perpendicular* is the vertical line drawn where the design waterline intercepts with the aft side of the rudder post. The mid point between aft and fore perpendicular is known as *midship*, (see Figure 2.6).

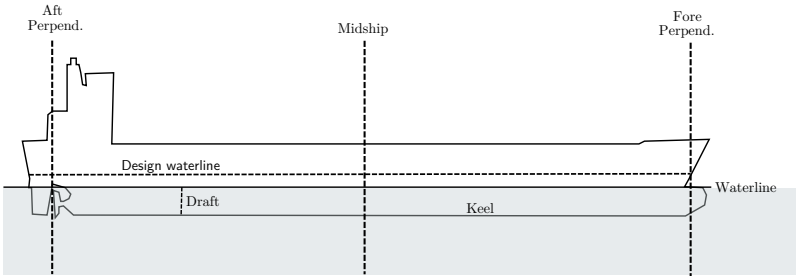


Figure 2.6: Position of perpendicular and midship in a container vessel

Draft is the distance between the *keel* (bottom of the vessel) and the water line, and can be calculated at any point of the vessel. When calculated at the perpendiculars it is called *Fore* and *Aft* draft, respectively. There are three requirements for the draft at the perpendiculars. First, the aft draft of a container vessel must be high enough for the propeller to be submerged under water. Second, when calling into a port, the maximum of aft and fore draft must be smaller than the allowed draft at the port to avoid the keel hitting the bottom. Third, the maximum draft must be high enough to allow port cranes to work properly.

The *trim* of a vessel is defined as the difference between its aft and fore draft. When the trim is positive the vessel is trimming by the stern, and when the trim is negative the vessel is trimming by the bow. Trim and draft play an important role in fuel consumption. Draft increases as a consequence of an increment in the vessel displacement. The heavier the vessel gets, the faster the engine must go to keep up the speed. With respect to trim, the vessel can be trimmed by the bow to achieve its *optimal trim*; that is, a trim where, due to the shape of the hull, sailing of the vessel will be as efficient as possible.

The *center of gravity* (G), of a container vessel is the point where the gravitational force may be taken to act. The *center of buoyancy* (B), is the center of gravity of the water displaced by the vessel, and it is where the buoyancy force can be taken to act. When the vessel is at *even keel*, as depicted in Figure 2.7 (a), G and B are aligned with the *center line*, a straight line midway of the vessel running from bow to stern. When an external force is applied to the vessel, e.g., wind or waves, it heels and the submerged section of the hull changes, moving the center of buoyancy in the same direction as the external force. The intersection of the center line of the vessel with a vertical line that goes through the new center of buoyancy (B_1), is the *metacenter* (M), (see Figure 2.7 (b)). Since G and B_1 are misaligned, the two forces act in opposite directions with the same magnitude (weight w on G and buoyancy force b on B_1), and generate the *moment of statical stability*, that is, the moment to return the vessel to its original position after

the external force has disappeared. The lever arm of the moment of statical stability is known as the *righting lever*, and it is the transversal distance between the two forces. The moment is calculated from G, and it is equal to b times the righting lever. It is assumed that there are no free bodies in the vessel, since they force G to move as well.

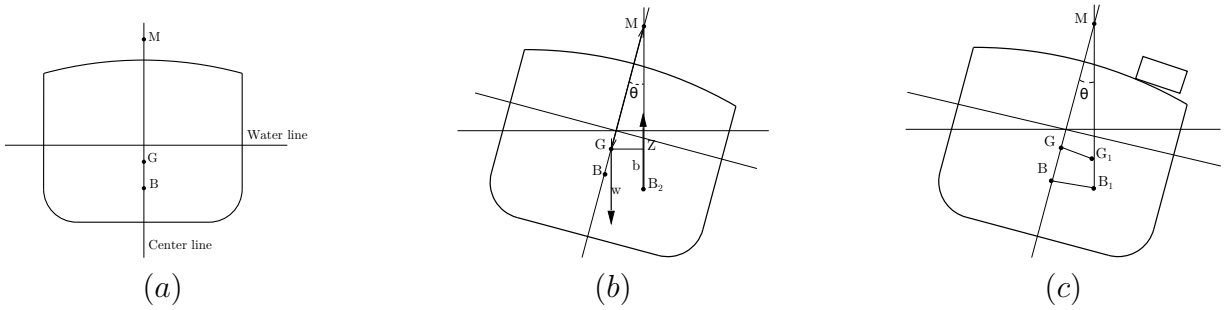


Figure 2.7: Transversal view of (a) Stable vessel. (b) Vessel perturbed by a external force (heeling). (c) Vessel perturbed by adding new weight (listing)

For heeling angles up to 15 degrees, we may consider that the buoyancy force acts vertically upwards to the initial metacenter. From figure 2.7 (b), we define Z to be the projection of G on B_1M with respect to the transverse axis. The righting lever GZ can then be calculated as $GMsin\,\theta$, where θ is the heeling angle, and GM the distance between the vertical coordinate of the metacenter and the center of gravity, also known as *metacentric height*. We can then see the direct relation between the moment of statical stability and the initial GM for small heeling angles, since as the GM increases, so does the moment. Vessels with large GM are stiff and resistant to rolling. Once they start rolling, however, they can achieve considerable speed when coming back to their original position due to the increased moment of stability. This is particularly inconvenient for crew and passengers, but also for container vessels that carry container stacks on deck. On the other hand, a decidedly small or negative GM is not convenient either, since the vessel will tend to capsize before it rolls back to its original position. Thus, for a vessel to be stable with respect to its statical stability, its GM must be within a safety range.

Now imagine a container vessel where the resultant forces acting on it are zero, (Figure 2.7 (a)). When a body already on the vessel is moved transversally, or a new body is added such that the center of gravity of the vessel moves transversally, the vessel *lists* (see figure 2.7 (c)), to bring its center of gravity transversally under the metacenter. Stability conditions dictate that a vessel must be at even keel to be seaworthy.

Two main forces constantly act on the structure of a vessel in opposite directions. One is the gravity force acting downwards. The other is acting upwards and is the buoyancy force, caused by the water displaced by the submerged part of the hull. These two forces, both with the same magnitude, are not uniformly distributed along the vessel. For instance, let us assume that a vessel is divided in longitudinal sections that are independent and can move vertically, as Figure 2.8 shows, and that each section

weights differently. Their weight depends on what has been loaded in the section and the weight of the vessel structure in the section. Sections with downward arrows weigh more than the buoyancy they provide, and sink deeper into the water until the amount of displaced water equals the section's weight. In sections with upward arrows, the weight is smaller than the buoyancy, and they raise until the amount of water displaced matches the section's weight. Though vessels are not divided into independent sections, Figure 2.8 illustrates how the weight and buoyancy forces are distributed along the vessel creating stress forces in the structure of the ship. Limits on these stress forces are given for a set of points along the vessel called *frames*.

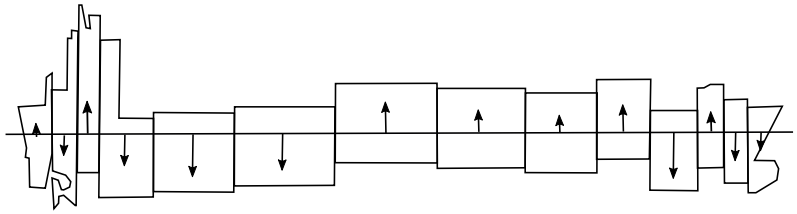
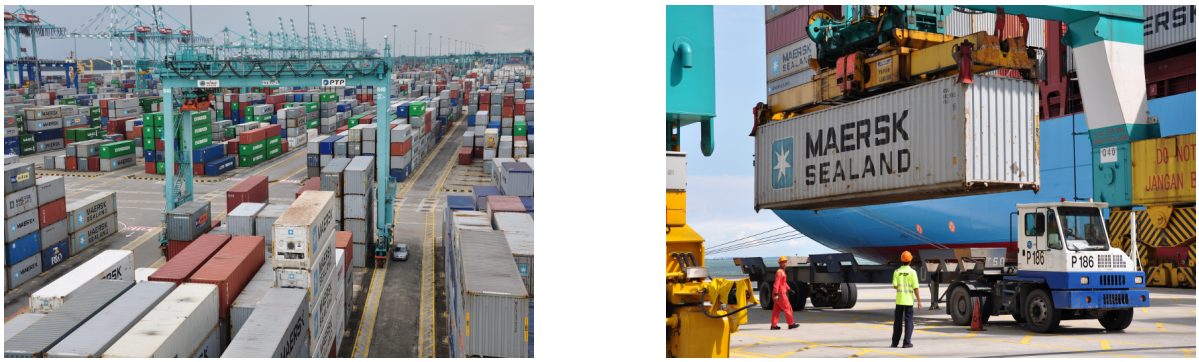


Figure 2.8: Example of the effects of uneven distribution of weights and buoyancy forces along different sections of a container vessel.

The *shear force* on a frame can be approximated by the sum of the resulting vertical forces acting on the vessel section aft or fore of the frame. The weight of a section is calculated by adding up the individual weights of all the objects in the section, including the weight of the vessel structure. The calculation of the buoyancy force, nevertheless, is more challenging. As previously stated, the buoyancy force is the weight of the water displaced by the submerged section of the hull. Thus, we can calculate the weight of the water displaced by calculating the volume of the submerged section when knowing the density of the water. This volume depends on the current draft of the vessel and the shape of the hull. Due to the irregular shape of hulls, the ship builder provides a set of reference points along the vessel called *stations*, where the immersed area of the transverse section of the hull is given as a function of draft by the *bonjean curve* of the station. For any two consecutive stations, we calculate the immersed volume between the two stations, and approximate the buoyancy force in that section by averaging the areas calculated at the current draft, times the distance between the two stations. The *bending moment*, that is, the moment forcing the vessel to bend at a specific point of the vessel, must be within limits at each frame. The bending moment can be approximated by multiplying each vertical force acting on the vessel section aft or fore of the frame, with the distance from the frame itself to where the force acts.

2.3 Maritime Container Terminals

Maritime container terminals are port sections where containers are transshipped between ships, and between ships and land vehicles. They are divided into three main sections: the *quayside*, the seaside section of the terminal where the vessels arrive and containers are loaded and unloaded; the *yard*, where containers are temporarily piled up on long rows of container stacks, as can be seen in Figure 2.9 (a); and the *landside*, where containers are picked up and dropped off by trucks and trains. Containers are moved between the different sections of the terminal by truck trailers, Automated Guided Vehicles (AGV), and straddlers, among others. Within the yard they are manipulated by *gantry cranes* mounted on rubber tiers (RTG), or rails. Figure 2.9 (a) shows an RTG moving over a row of containers lying in a terminal yard. On the quayside, *quay cranes* with *spreaders* carry out the loading and unloading of containers from the vessel, accessing only the top-most containers in stacks. Figure 2.9 (b) depicts a container being unload from a container vessel by a quay crane and loaded onto a truck. The figure shows how a spreader holds a container by its top four corners. Quay cranes



(*a*) (*b*)

Figure 2.9: (a) Terminal yard with a front view of a gantry crane. (b) Container being unloaded from a vessel to a truck by a quay crane in the quayside of a terminal.

can perform between 30-37 *moves* (loading or unloading of a container), per hour. Most of the modern quay cranes can do twin-lifting, that is, lift two *20'* containers at the same time, increasing their average moves per hour to around 25 for *20'*, i.e., lifting 25 pairs of *20'* containers per hour, and 30 for *40'* containers. Quay cranes are also used to lift hatch covers, an operation that takes longer since spreaders need to be aligned for the hatch. The same crane can be assigned to attend multiple bays, and a *crane-set*, the operation of changing the crane position, can take around two minutes. Cranes run on rails that are lying parallel to the vessel and cannot pass each other. There must be a safety distance, e.g., 40, 60, 80 feet, between two adjacent cranes when they are working in parallel.

The partition of vessel sections into areas handled by a single crane is known as *crane split*. A good crane split minimizes crane makespan, which directly impacts port

fees since the carrier has to pay not only per container move but also per hour of equipment usage.

2.4 Container Stowage

Containers must be loaded on board the ship to reach their destination. This process is not trivial since containers cannot be freely stowed in the vessel. For a start, containers must be distributed along the vessel such that seaworthiness is not compromised, (see section 2.2.2). A vessel that is not seaworthy will not be allowed to leave the port. Moreover, low level stacking constraints, i.e., container stack capacity limits (weight and height), that are specified in a *vessel profile* describing the vessel layout, and stacking rules based on container features (e.g., length, IMO types, Pallet-wide types, etc.), also come into play. Containers are provided with corner castings that are used to secure them to the vessel and to other containers. Each of the four bottom corner castings of a container must rest on the corner castings of other containers or sockets of the vessel. This implies that *20'* containers cannot be stowed on top of containers of greater length, since they do not have castings in the middle. Extra castings and frames at approximately 75cm from the back and front door have been installed on *45'* containers. This allows *45'* containers to be stowed on top of *40'* containers. It seems to be possible as well to do the opposite, stow *40'* on top of *45'* containers, but this is not allowed. Containers with special features also follow specific stacking rules. For instance, there is a special type of pallet-wide container (*PW6*), that cannot be stowed side by side in consecutive stacks, and another type (*PW3*), that requires a normal container be stowed between two stacks stowing this type of pallet-wide container. In the case of *IMO* containers, some of the IMO groups have proximity constraints with respect to other IMOs. IMO-2 requires one container of separation to any other IMO-2 container, while IMO-3 and IMO-4 have more complex rules.

Stacking rules also depend on the section of the vessel. As described in section 2.2.1, container vessels are divided into on deck and below deck sections. The holds are provided with cell guides at the aft and fore part of the bay. Containers are stowed and locked on top of each other by *stacking cones* that mainly resist vertical forces, but are strong enough to provide adequate longitudinal and transverse support below deck. It is important to notice that containers cannot hang in the air, they have to be stowed at the bottom of a bay or on top of other containers. Odd cells at the bottom of a stack must be filled up with *20'* containers before using the cells on top. It is not possible to stow single *20'* containers on top of stacks since the inward end is not secure. Container vessels of previous generations had special holds for *45'* containers, but nowadays all *45'* are stowed on deck.

In the on deck section, *twistlocks*, connectors designed to withhold stronger vertical and horizontal forces, replace stacking cones to secure containers to each other or to the deck. Up to the second tier, twistlocks provide enough support for the containers.

Beyond the second tier, and up to the fifth, *lashing rods* are needed to provide extra support to containers stowed in this area. The rods are tied to the deck and need to be kept tight all times. They are constantly adjusted during the journey. Containers *45'* long are stowed on deck. They can be stowed on top of *40'* containers but the opposite is not possible, since the *40'* containers on top will not be accessible for lashing. To go above the fifth tier in stacks on deck containers must be tied to the lashing bridge. This metallic structure goes a few tiers up from the deck and allow lashing rods to go further up in the container stack, (up to the bottom corner castings of containers in the fifth tier). The length between lashing bridges is equivalent to the size of a *40'* container, thus, *45'* containers must be stowed above the lashing bridge. In the case of *20'* containers, it is only possible to secure one side of the container to the lashing bridge. The other side, facing inwards, is lashed to the deck up to the fifth tier. After the fifth tier, the containers are only secured by twistlocks. This makes pure *20'* stacks weaker.

The lashing rods of *20'* containers facing inwards must always be accessible during the journey. Since it is not possible to walk between containers from one lashing bridge to another, *40'* containers in *mixed bays* (bays that hold *20'* and *40'* containers), must be stowed in such a way that there is an access path either from port or starboard, to the *20'* containers. Figure 2.10 (a) shows a stowage pattern in a mixed bay with an access path to the *20'* containers, while Figure 2.10 (b) depicts a pattern where *40'* containers block access to the *20'* containers. As a consequence of the need for this

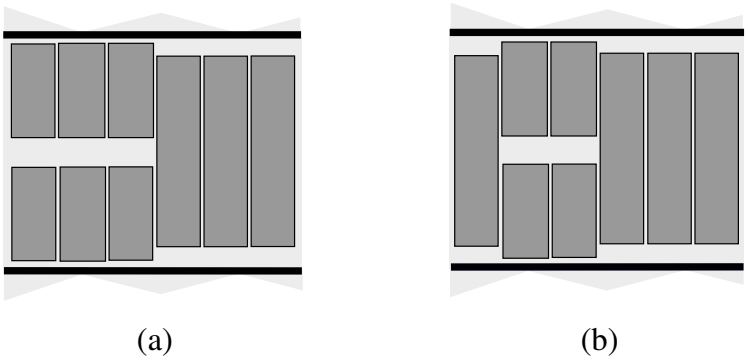


Figure 2.10: Small bay seen from above. (a) Stowage where there is free access to *20'* lashing rods. (b) Stowage *40'* containers blocking access to *20'* lashing rods.

access path, *20'* and *40'* containers must rest on different outer *sockets* (structures on the hatch covers where containers on the bottom layer are positioned), as depicted in Figure 2.11. The area occupied by two *20'* containers is no longer the same as that of a single *40'* container, thus, *Russian stacks* (stacks where *20'* and *40'* containers are stowed together), are not allowed on deck. In newer vessels, however, it is possible to do Russian stacks on deck.

Other factors such as the stability of the vessel, neighbor stacks, and visibility must also be considered while stowing container stacks on deck. When the GM forces the

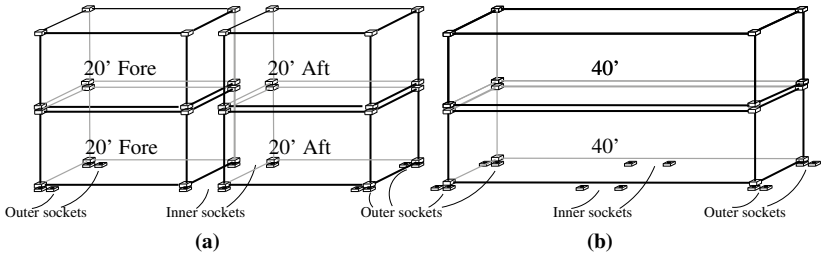


Figure 2.11: The position of the sockets on a bay where lashing rods of *20'* containers need to be accessed. (a) Shows *20'* containers using inner and outer sockets. (b) Shows *40'* containers using only the outer sockets. Notice that *20'* and *40'* containers use a different set of outer sockets.

vessel to have a short rolling period (stiff vessel), containers at the top tiers experience strong forces that put considerable stress on the twistlocks and the structure of the containers in the lower tiers of the stack. As a consequence, the size of container stacks on deck must be reduced to avoid their collapse. The impact of this problem, however, can be considerably reduced by: 1) Lowering the GM of the vessel and, 2) Redistributing the weights in the container stacks by stowing the heavy containers at the bottom tiers and the light ones at the top.

Container stacks on deck must allow full visibility from the pilot house. An imaginary line is drawn from the pilot house to a point 500m or twice the length of the vessel, whichever is the smallest, measured from the foremost point of the vessel (*line of sight*), (see Figure 2.3). Essentially, all containers on deck must be below this line. It is important to notice that the line of sight is influenced by the fore draft of the vessel. The longer the fore draft, the less restrictive the line of sight becomes, increasing the on deck capacity of the vessel.

2.5 Stowage Plans

Stowage plans assign containers to vessel slots. These plans are made by *stowage coordinators* who work for the liner shipping company before the vessel calls into port. In some liner shipping companies the time a stowage coordinator has to produce a stowage plan is six hours. When making a stowage plan, stowage coordinators have available a *loadlist* of containers (i.e., the list of containers to load in the current port), the vessel condition when it calls the port, port information, and a forecast based on historical data. The *loadlist* of containers includes detailed information about the containers, such as weight, height, length, type (e.g., standard, reefer, IMO, Pallet-wide), and discharge port. The *release* is a list of containers onboard the vessel when it calls into port, it has the same container information as the loadlist and includes the exact position where containers are stowed. The status of the ballast, fuel, diesel, and oil tanks when the vessel calls into port is also provided to the stowage coordinators together with

information on changes to any of the tanks and non-cargo loads that the vessel carries.

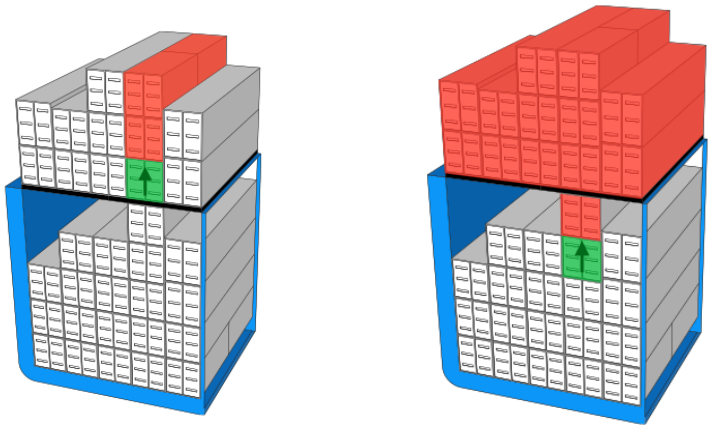
Stowage coordinators are provided with port information for the current and downstream ports. This information includes the number of quay cranes assigned to the vessel and the maximum draft at each port. Stowage plans are made for the current port but stowage coordinators must take downstream ports in the vessel schedule into account, since the decisions they make in the current port affect future stowage plans. Considering that forecasts are not always accurate, stowage coordinators follow rules-of-thumb to make robust stowage plans, e.g., avoid stowing non-reefer containers into reefer slots, keep the number of different discharge ports in the same location as low as possible, and cluster together containers with the same discharge ports that are stowed in the same vessel sections.

Once a vessel is loaded with a stowage plan the vessel must be seaworthy, container stacks must satisfy all stacking rules and limits, and containers with special requirements must have been handled correctly. Feasible stowage plans, however, should be economically viable as well. They must save time at port and, if possible, reduce fuel consumption. There are two main features of a stowage plan that impact the time a vessel spends at port. The first is the distribution of moves over the vessel. Stowage coordinators attempt to distribute evenly the moves along the vessel to minimize the crane makespan. Figure 2.12 (a) presents a distribution of containers with a high makespan, where moves concentrate at the stern of the vessel. A better distribution of the moves along the vessel, as Figure 2.12 (b) depicts, reduces the makespan.

Figure 2.12: Crane split of three cranes assigned to a container vessel. The numbers in each vessel section indicates the moves that need to be performed there. (a) Bad distribution of moves along the vessel with a crane makespan of 200. (b) Improved distribution distribution of moves with a crane makespan of 130.

The second feature is the number of unnecessary moves performed to unload the containers on the vessel. Let us assume that we have two different sets of containers, A and B, stowed in the same bay. The set of A containers is going to Shanghai and is stowed right on top of a set of B containers going to Yokohama. Since containers can only be accessed from the top, all containers in A must be unloaded to access containers in B. If the container vessel calls Shanghai before calling Yokohama, containers in A will be unloaded in Shanghai, allowing free access to the containers in B at Yokohama. On the other hand, if the vessel calls Yokohama first, the containers in A will *overstow* the containers in B and will have to be *shifted*, (unloaded in Yokohama and moved back into the vessel to the same or a different position). Overstowage is measured by the number of containers that have to be shifted. In our previous example it is the

number of containers in A. There are two types of overstowage, *stack overstowage*, where containers in sets A and B are stowed in the same stack (see Figure 2.13 (a)), and *hatch overstowage*, where containers in sets A and B are stowed above and below the same hatch cover, respectively, (see Figure 2.13 (b)).



(a) (b)

Figure 2.13: Examples of different kinds of overtoswage in a bay section. The green container is going to Yokohama, the red containers are going to Shanghai. The vessel calls Yokohama first. (a) *Stack overstowage*: Containers on top of the green container must be shifted in Yokohama. (b) *Hatch overstowage*: Containers on top of the green container, few from the same section and all from the section on top, must be shifted in Yokohama.

With respect to fuel saving, stowage coordinators aim at distributing containers along the vessel such that the trim gets as close to optimal as possible.

Chapter 3

Slot Planning

In chapter 1, we describe a 2-phase decomposition approach for scalable stowage planning. In this approach, the master planning phase assigns containers to locations along the vessel, satisfying high-level constraints and objectives. The slot planning phase, on the other hand, stows containers into vessel slots, satisfying low level stacking constraints and objectives, following the assignment made during the master planning phase. This chapter introduces the definition of a representative problem for slot planning locations, together with two optimization models to solve this problem efficiently. This chapter is based on Delgado et al. (2009), and Delgado et al. (2012).

3.1 Introduction

Slot planning is the second phase of the 2-phase heuristic decomposition depicted in Figure 1.3. In this phase, the containers assigned to locations during master planning are stowed into vessel slots. Each location is stowed independently such that container stacks fulfill stacking rules and satisfy capacity limits, and optimize objectives that focus on robustness. Figure 3.1 depicts the layout of the slot planning phase considered in this chapter. A typical large container vessel has about 100 locations, which im-

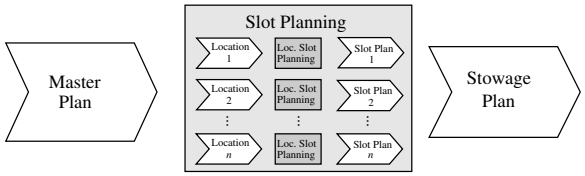


Figure 3.1: Layout of slot planning phase.

plies that the slot planning phase solves about 100 independent slot planning problems. Thus, when given at most 10 minutes[1] to generate complete stowage plans, including

[1]Time limit set by our industrial collaborator for the generation of a stowage plan.

master planning, on hardware that does not support heavy parallelization, we aim at solving each slot planning problem in less than one second. This is not trivial since slot planning is NP-hard and each location may hold up to several hundred containers. Real slot planning problems are full of details, including a wide variety of special vessel structures and container types. To make their study practical, we introduce the Container Stowage Problem for Below Deck Locations (*CSPBDL*), a representative model for stowing containers in locations below deck formulated together with our industrial collaborator. Even though this is a simplified representation of the problem, it is, to our knowledge the most detailed model published to date. We formulate an Integer Programming (IP) and Constraint Programming (CP) model for solving the *CSPBDL* to optimality.[2] The CP model uses state-of-the-art modeling techniques including multiple viewpoints, specific domain pruning rules, and dynamic lower bounds. The IP model is a 0-1 formulation where cuts are introduced to strengthen the LP relaxation.

Despite the size and NP-hardness of slot planning problems, our computational results show that they can often be solved very quickly in practice. We have devised two experimental setups to evaluate our models. In the first setup, we have derived 236 test instances from real stowage plans from a system deployed by our industrial collaborator, (Guilbert and Paquin (2010)). 92% of the instances are solved using a state-of-the-art constraint solver on our CP model within one second. Similar but slightly worse results were obtained with the IP model. In the second setup, we incorporate our CP model, the most successful model based on the first set of experiments, into a module that computes stowage plans for container vessels, (Pacino et al. (2011)). We used this module to stow 18 industrial stowage conditions. In this setup, our CP model managed to generate optimal slot plans for up to 94.8% of the locations with containers to stow. Thus, somewhat to our surprise, it is possible to define optimal models of slot planning problems that can be solved fast enough to be used in stowage planning optimization tools.

This chapter is organized as follows. Section 3.2 defines the problem we address in this chapter and related work is presented in Section 3.3. In Section 3.4, we give a detailed description of our IP model. Section 3.5 gives a brief introduction to global constraint modeling, and in Section 3.6 we present our CP model. Computational results are presented in Section 3.7 and conclusions and directions for future work are discussed in Section 3.8.

3.2 Problem Statement

In this chapter we study the slot planning problem, which is to assign a set of containers to slots in a location that may already hold release containers. Due to the large number of constraints and objectives involved in real slot planning, we have developed a representative version of the problem for below deck locations called the *CSPBDL*. On

[2]A column generation model was also formulated for this problem and experimentally tested in Delgado et al. (2009).

deck locations share most constraints and objectives with below deck locations, thus we expect similar computational results for them. The *CSPBDL* covers all constraint and objective classes of the problem, and we anticipate a high correlation with a complete problem model in terms of solution algorithm performance. Specifically, the *CSPBDL* includes stacking rules for *20'* and *40'* containers, FEU and TEU stack overlapping, reefer containers, release containers, and weight and height constraints. The objectives include overstowage and three rules of thumb used by stowage coordinators to ease the stowing of containers in downstream ports. We limit the containers we consider to be *20'* and *40'* long, 8'6" and 9'6" high, and reefer and non-reefer. A feasible *CSPBDL* must satisfy the following rules.

a) Assigned cells must form stacks, (containers stand on top of each other in the stacks. They must have support from below)
b) *20'* containers cannot be stacked on top of *40'* containers
c) A *20'* reefer container must be placed in a reefer slot. A *40'* reefer container must be placed in a cell with at least one reefer slot
d) The length constraint of a cell must be satisfied, (some cells only hold *40'* or *20'* containers)
e) The sum of the heights and weights of the containers stowed in a stack are within the stack limits
f) All release containers must be stowed in their original slots and they cannot be swapped to any other slots
g) A cell must be either empty or with both slots occupied

Additionally, an optimal *CSPBDL* minimizes the sum of the following objectives, listed in order of importance:

h) Minimize overstows. A penalty is paid for each container overstowing any containers below
i) Avoid stacks where containers have many different Port of Discharge (POD). A penalty is paid for each POD included in a stack
j) Keep stacks empty if possible. A penalty is paid for each stack used
k) Avoid loading non-reefer containers into reefer slots. A penalty is paid for each non-reefer container stowed in a reefer slot

The second, third, and fourth objectives are rules of thumb of the shipping industry when generating slot plans for downstream ports in the route of a vessel. Using as few stacks as possible increases the available space in a location and reduces the possibility of overstowage in future ports, so does clustering containers with the same POD. Minimizing the reefer objective allows more reefer containers to be loaded in future ports. The cost units reflect the importance of each objective and has been defined by our industrial collaborator.

The CSPBDL is NP-Hard. We show that the bin-packing problem can be reduced to the *CSPBDL*. All items in a bin-packing problem are defined as *40'*, standard height and non-reefer containers with the same POD, which sizes are the weight of the containers. Bins are defined as stacks where height limits are set to be sufficiently large to be non-restrictive, with no reefer slots, and which sizes are the weight capacity of the stacks. An optimal solution to an arbitrary bin-packing problem can be found by finding an optimal solution to the *CSPBDL*. When all containers have the same POD and are non-reefer, objectives i and j dominate the objective function and the number of stacks used is minimized, showing that the *CSPBDL* is NP-Hard.[3]

3.3 Literature Review

Slot planning optimization algorithms have either been studied as embedded into single-phase models, or as a part of multi-phase decompositions for generating stowage plans. In the first category, Avriel et al. (1998), Dubrovsky et al. (2002), and Ambrosino and Sciomachen (1998), propose simple models to stow complete vessels that address similar problems to slot planning optimization. Avriel et al. introduce a 0-1 IP model and a heuristic called the *suspensory heuristic*, to stow vessels described as a collection of columns and rows, (a rectangular bay). They consider all containers to have the same features and focus on minimizing re-shifting of containers, (overstowage). Dubrovsky et al. presents a genetic algorithm model with the same assumptions as Avriel et al.'s. They claim, however, that their approach is flexible enough to include new constraints. Ambrosino and Sciomachen present a CSP model. Though this model is meant to stow a complete vessel, inter-bay stability constraints can be dropped in order to resemble the slot planning problem. This approach considers *20'* and *40'* containers, but lacks reefer and high-cube containers. Their objective is to minimize overstowage and maximize the number of containers loaded. Aslidis (1984), Botter and Brinati (1992), Sciomachen and Tanfani (2003), and Li et al. (2008), present more complex models that also include several of the constraints considered in the master planning problem. Aslidis introduces stacking heuristics for minimizing overstowage, while Botter and Brinati, and Li et al. present 0-1 IP models. Botter and Brinati also present two heuristics to stow containers since their IP model is not scalable to real-life instances. Sciomachen and Tanfani introduce a heuristic approach based on the 3D-packing problem. These approaches consider *20'* and *40'* containers and overstowage minimization. Sciomachen and Tanfani also consider high-cube containers.

In the second category, slot planning problems are solved in connection with multi-phase approaches for complete vessel stowage planning. Wilson and Roach (2000), briefly describe a tabu search algorithm for solving a version of slot planning that must have included reefer slots, length restrictions, minimized overstowage, and avoided POD mixing of stacks. They claim that near optimal solutions could be computed fast, but

[3]It can be shown that overstow minimization is also an NP-hard component of the *CSPBDL*, but the proof requires a version of the problem with uncapacitated stacks Avriel et al. (2000).

only experimental results for generating a complete stowage plan for a single vessel are described. Kang and Kim (2002), describe an enumeration approach for solving a very simple version of slot planning, where only overstow minimization and sorting of *40'* containers after weight are considered. As for Wilson and Roach, no independent experimental evaluation of the algorithm is provided. Ambrosino et al. (2009), describes a 0-1 IP model for optimally stowing subsets of vessel bays holding containers with the same POD. The model minimizes the time for stowing containers. *20'* and *40'* containers are considered, and containers are sorted according to weight in each stack. In the experimental section, complete stowage plans for a 198 and 2124 TEUs container vessels are generated. The maximum bay size is 20 TEUs for the small vessel and 120 TEUs for the big one. No computational time, however, is provided for solving these sub-problems. In a later work, Ambrosino et al. (2010), presents a constructive heuristic to solve the same sub-problem as the one described in Ambrosino et al. (2009), as part of a complete approach to stow vessels. Using this heuristic, they are able to stow a vessel of 5632 TEUs. The heuristic uses 11.8 seconds on average to stow all the bays, but the physical layout of the vessel is not described in detail. Zhang et al. (2005), and Yoke et al. (2009), present multi-phase approaches where the problems solved during the slot planning phase are not independent of each other.

A deployed industrial system introduced by Guilbert and Paquin (2010), that provides data to our experiments, solves slot planning problems as linear assignment problems with side constraints. Their model considers all containers and most of the constraints and objectives present in the *CSPBDL*. Overstowage is only considered with respect to loaded containers, and though they minimize mixed stacks with *20'* and *40'* containers, constraint (b) is not present.

3.4 The IP Model

In this section, we introduce a binary IP model formulated to solve the *CSPBDL*. We remind the reader that the sets and constants used in the models presented in this chapter have been previously defined in the section *Sets and Constants*, at the beginning of this document. Below we present the variables used in this model.

Decision Variables

$c_{jki} \in \{0,1\}$ Container i being stowed in cell k, stack j.

Auxiliary Variables

$o_i \in \{0,1\}$ Container i overstowing.
$d_{jp} \in \{0,1\}$ At least one container in stack j being unloaded at pd.
$e_j \in \{0,1\}$ Stack j being used.
$\delta_{jkp} \in \{0,1\}$ Container below cell k, stack j being unloaded before port p.

The sets of variables o, p, and e are used for computing the cost of overstow (h), clustering (i), and using stacks (j) according to the *CSPBDL*. The variables in c rep-

resent the slot plan. The set of auxiliary variables *delta* represents indicator variables introduced to model the overstowage objective. The IP model is defined as:

$$\begin{aligned} min \quad & C^v \sum_{i \in Q} o_i + C^p \sum_{j \in J} \sum_{p \in P - \{1\}} d_{jp} + C^u \sum_{j \in J} e_j \\ & + C^r \sum_{j \in J} \sum_{k \in K_j} \Big(R_{jk} \sum_{i \in Q^{40}} c_{jki}(1 - R_i^C) + \sum_{i \in Q^{20}} c_{jki}(\frac{1}{2} R_{jk} - R_i^C) \Big) \end{aligned} \tag{3.1}$$

s.t.

$$\frac{1}{2} \sum_{i \in Q^{20}} c_{j(k-1)i} + \sum_{i \in Q^{40}} c_{j(k-1)i} - \sum_{i \in Q^{40}} c_{jki} \geq 0 \qquad \forall j \in J,\, k \in K_j - \{1\} \tag{3.2}$$

$$\sum_{i \in Q^{20}} c_{jki} - \sum_{i \in Q^{20}} c_{j(k-1)i} \leq 0 \qquad \forall j \in J,\, k \in K_j - \{1\} \tag{3.3}$$

$$\frac{1}{2} \sum_{i \in Q^{20}} c_{jki} + \sum_{i \in Q^{40}} c_{jki} \leq 1 \qquad \forall j \in J,\, k \in K_j \tag{3.4}$$

$$\sum_{j \in J} \sum_{k \in K_j} c_{jki} = 1 \qquad \forall i \in Q \tag{3.5}$$

$$\sum_{i' \in Q^{20}} c_{jki'} - 2c_{jki} \geq 0 \qquad \forall j \in J,\, k \in K_j, i \in Q^{20} \tag{3.6}$$

$$\sum_{i \in Q} R_i^C c_{jki} - R_{jk} \leq 0 \qquad \forall j \in J,\, k \in K_j \tag{3.7}$$

$$\sum_{k \in K_j} \sum_{i \in Q} W_i^C c_{jki} \leq W_j^S \qquad \forall j \in J \tag{3.8}$$

$$\sum_{k \in K_j} (\frac{1}{2} \sum_{i \in Q^{20}} H_i^C c_{jki} + \sum_{i \in Q^{40}} H_i^C c_{jki}) \leq H_j^S \qquad \forall j \in J \tag{3.9}$$

$$\sum_{k'=1}^{k-1} \sum_{p'=2}^{p-1} \sum_{i \in Q} A_{ip'} c_{jk'i} - 2(k-1)\delta_{jkp} \leq 0 \qquad \forall j \in J,\, k \in K_j,\, p \in P \tag{3.10}$$

$$A_{ip} c_{jki} + \delta_{jkp} - o_i \leq 1 \qquad \forall j \in J,\, k \in K_j,\; p \in P,\, i \in Q \tag{3.11}$$

$$e_j - c_{jki} \geq 0 \qquad \forall j \in J,\, k \in K_j,\, i \in Q \tag{3.12}$$

$$d_{jp} - A_{ip} c_{jki} \geq 0 \qquad \forall j \in J\,, k \in K_j,\; p \in P,\, i \in Q \tag{3.13}$$

$$c_{jki} = 1 \qquad \forall (j,k,i) \in M \tag{3.14}$$

The objective function (3.1) is a weighted sum of the four objectives as defined in the *CSPBDL*. The first three objectives are calculated straightforward since there are specific auxiliary variables in the model that account for them. The fourth objective

is calculated by determining the number of non-reefer containers stowed in slots with reefer plugs. *40'* and *20'* containers are considered independently.

Inequality (3.2) ensures that there is either two *20'* or one *40'* containers below a cell stowing a *40'* container, while inequality (3.3) constrains the containers below a cell stowing *20'* containers to be *20'* long (b). Inequality (3.4) requires that all cells stow at most either two *20'* or one *40'* container. Containers are forced to be stowed in exactly one cell by (3.5). Inequality (3.6) forces the number of *20'* containers in a cell to be 0 or 2, since the two sides of a stack must be synchronized (g). The reefer capacity of a cell is constrained by inequality (3.7), covering the fact that all reefer containers in a cell must be provided with a reefer plug each (c). The weight and height limits of stacks (e) are enforced by (3.8) and (3.9), respectively. Inequality (3.10) ensures that the variables δ_{jkd} are assigned the correct value according to their semantics. These variables are then used in inequality (3.11) to assign the overstowage variables o_i for each container. Inequality (3.12) sets a lower bound for the variable related to the empty stack objective (j) for each stack, and inequality (3.13) does the same for the variables related to the clustering objective (i) for each stack at each POD. Release containers are assigned to their corresponding cell by equality (3.14).

Cuts

We add cuts that focus on removing solutions with non-integer values assigned to variables δ_{jkp} by the Linear Programming (LP) relaxation. First we decompose inequality (3.10) into several inequalities, one for each variable $c_{jk'i}$, that combined together are semantically equivalent to (3.10):

$$A_{ip'}c_{jk'i} \leq \delta_{jkp} \qquad \forall j \in J,\, k \in K_j,\, p \in P,\, i \in Q,\, k' \in K',\, p' \in P' \tag{3.15}$$

where $K' = \{k' | k' \in \{1, ..., k-1\}\}$ and $P' = \{p' | p' \in \{2, ..., p-1\}\}$. We then increase the size of the left hand side term of (3.15) by considering at once all containers unloaded earlier than port p. Two inequalities are introduced to the model (3.16, 3.17), since *20'* and *40'* containers need to be treated differently. Additionally, cut (3.18) adds terms to the left hand side of (3.15) by considering all cells below cell k. The cuts are defined by:

$$\frac{1}{2} \sum_{i \in Q_p'^{20}} c_{jk'i} \leq \delta_{jkp} \qquad \forall j \in J,\, k \in K_j,\, p \in P,\, k' \in K' \tag{3.16}$$

$$\sum_{i \in Q_p'^{40}} c_{jk'i} \leq \delta_{jkp} \qquad \forall j \in J,\, k \in K_j,\, p \in P,\, k' \in K' \tag{3.17}$$

$$\sum_{k'=1}^{k-1} A_{ip'}c_{jk'i} \leq \delta_{jkp} \qquad \forall j \in J,\, k \in K_j,\, i \in Q,\, p \in P,\, p' \in P' \tag{3.18}$$

where $Q_p'^{20}$ and $Q_p'^{40}$ are the set of *20'* and *40'* containers with POD earlier than p, respectively.

3.5 Global constraint modeling

A Constraint Satisfaction Problem (CSP) is a triple (X, D, C) where X is a set of variables, D is a mapping of variables to finite sets of integer values, and $D(x)$ representing the domain of $x \in X$ and $D(X) = \Pi_{x \in X} D(x)$ being the Cartesian product of domains, and C is a set of constraints. Each $c \in C$ is defined over a sequence $X' \subseteq X$ as a subset of allowed combinations of $D(X')$. A solution to a CSP is a complete assignment that maps every variable to a value from its domain that satisfies all constraints in C.

Constraint Programming (CP) is a relatively new technique that combines local consistency algorithms with search. The process of removing inconsistent values from the domain of the variables is called *propagation*. A depth-first backtracking search explores the search space of the problem incrementally. It extends a *partial solution* by selecting unassigned variables from X and assigning them to values from their domains. This selection process is called *branching*, and a strategy to select variables and values following a specific criteria is called a *branching strategy*. Propagation is executed every time new branching is generated. If the domain of each variable has been reduced to a single value, the CP solver has found a solution to the CSP. For a partial solution, we refer to the minimum and maximum value of the domain of variable x as $\underline{x}$ and $\overline{x}$, respectively. A cost function is defined for a CSP in order to evaluate the quality of its solutions and *branch and bound* is used to find optimal solutions.

Constraints in CP share information through the variables in X. Each constraint has a scope $X' \subseteq X$, that often is relatively small compared to the size of X, limiting its propagation power. *Global constraints* have been introduced to overcome this. A global constraint groups together a set of small constraints capturing tractable structures for global propagation. Below is a brief description of the global constraints used in our CP model.

Let x be an integer variable, y a variable with finite domain, and $C = \{c_1, ..., c_n\}$ a set of constants. The *element constraint* Van Hentenryck and Carrillon (1988) states that y is equal to the x-th constant in C.

$$element(x, y, C) = \{(e, f) \,|\, e \in D(x), f \in D(y), f = c_e\}.$$

Let M be a deterministic finite automaton or a regular expression recognizing the strings in the language $L(M) \subseteq \Sigma^*$, and let $X = \{x_1, ..., x_n\}$ be a set of variables with $D(x_i) \subseteq \Sigma$ for $1 \leq i \leq n$. Then the *regular constraint* Pesant (2004), is defined as

$$regular(X, M) = \{(d_1, \ldots, d_n) \,|\, \forall i \,.\, d_i \in D(x_i), d_1 \cdots d_n \in L(M)\}.$$

Let n and v be two integer values, and $X = \{x_1, ..., x_m\}$ a set of finite domain variables. The *exactly constraint* ensures that exactly n variables in X are assigned to value v.

$$exactly(n, X, v) = \{(d_1, \ldots, d_m) \,|\, \forall i \,.\, d_i \in D(x_i), |\{d_i \,|\, d_i = v\}| = n\}$$

Let $X = \{x_1, ..., x_n\}$ and $Y = \{y_1, ..., y_n\}$ be two sets of finite domain variables with domains $D(X) = D(Y) = \{1, ..., n\}$. The *channeling constraint* states that a value j

assigned to a variable $x_i \in X$ represents the index of the variable $y_j \in Y$ that has been assigned value i from its domain. Formally

$$channeling(X, Y) = \{(e_1, \ldots, e_n, f_1, \ldots, f_n) | \\ \forall i, j \,.\, e_i \in D(x_i), f_j \in D(y_j), e_i = j \Leftrightarrow f_j = i\}.$$

A channeling constraint is used to increase the propagation power of the model by connecting several isomorphic variable sets, (also known as *viewpoints* Smith (2006)). If X is a set of variables representing positions with boxes $\{1, \ldots, n\}$ as domains and Y is a set of variables representing boxes with positions $\{1, \ldots, n\}$ as domains, then clearly a channeling constraint will link them consistently together. In particular, the channeling constraint embeds the *alldifferent constraint*, that in our example ensures that a position only can hold one box and vice versa.

3.6 The CP Model

In our CP model, the stack in the left most part of the location has the lowest index in J. Indices in S are assigned to physical slots as follows. For each cell, the aft and fore slots have consecutive indices. The slot indices in each stack are ordered bottom-up and the slot indices between stacks are ordered from left to right in the location. We have $S_k = S_k^F \cup S_k^A$, and $P_i^c < P_j^c$ iff the vessel calls the POD of container i before the POD of container j. Below we introduce the variables of our cp model.

	Decision Variables
$Conts = \{c_1, ..., c_{\|Q\|}\}$	$c_i \in S$, slot index of container i.
$Slots = \{s_1, ..., s_{\|S\|}\}$	$s_j \in Q$, container index of slot j.
	Auxiliary Variables
$Lengths = \{l_1, ..., l_{\|S\|}\}$	$l_j \in L^C$, length of container stowed in slot j.
$Heights = \{h_1, ..., h_{\|S\|}\}$	$h_j \in H^C$, height of container stowed in slot j.
$Weights = \{w_1, ..., w_{\|S\|}\}$	$w_j \in W^C$, weight of container stowed in slot j.
$Ports = \{p_1, ..., p_{\|S\|}\}$	$p_j \in P^C$, POD of container stowed in slot j.
$HS = \{hs_1, \ldots, hs_{\|J\|}\}$	$hs_j \in \{0, ..., H_j^S\}$, current height of stack j.
$o^v \in \{0, ..., \|Q\|\}$	Number of overstowing containers.
$o^u \in \{1, ..., \|J\|\}$	Number of used stacks.
$o^p \in \{1, ..., \|J\|\|P\|\}$	Number of different POD in each stack.
$o^r \in \{0, ..., \|S^R\|\}$	Number of non-reefers stowed in reefer cells.
$o \in \mathbb{N}$	Solution cost variable.
$Conts^V \subset Conts$	Virtual containers.
$Slots_i^E \subset Slots$	Slots with the same features in stack i.
$Lengths_i^\pi \subset Lengths$	Length variables of slots in the aft or fore part of stack i.

The decision variables represent the slot plan for a set of containers to be stowed. We use two isomorphic representations. The first one defines a decision variable for each container in Q to be stowed, and as domain of the variables the slots in S. The second one defines a decision variable for each slot in S, and as domain of the variables, the set of containers in Q to be stowed.

The two sets of decision variables mentioned above define two different viewpoints in our CP model. These two viewpoints are linked with a channeling constraint. The current formulation of the problem, however, does not allow a straightforward use of this constraint since in most of the cases the number of slots is larger than the number of containers, which breaks an important precondition of the channeling constraint. To tackle this issue, we modify the original definition of the problem by extending the number of containers with artificial containers to match the number of slots. First, since a *40'* container occupies two slots, all *40'* containers are split in two parts, *Aft40* and *Fore40*, with the size of a single slot. All *40'* containers from Q and Q^{40} are replaced by *Aft40* and *Fore40* containers. *Virtual containers*, Q^V, that will be stowed in slots meant to remain empty are also added. In the remainder of the chapter, Q will refer to this extended set of containers. Finally, $Q^{\neg R}$ is the set of non-reefer containers excluding virtual containers.

In addition to the two sets of decision variables, extra sets of auxiliary variables are defined to facilitate the modeling of the constraints and objectives. The objectives of the CP model are given by:

$$o^v = \sum_{i \in S^A} ov(i) \tag{3.19}$$

$$o^u = \sum_{j \in J} \left(\sum_{i \in S_j} p_i > 0 \right) \tag{3.20}$$

$$o^p = \sum_{j \in J} \left(\sum_{\rho \in P} \sum_{i \in S_j} (p_i = \rho) \right) > 0 \tag{3.21}$$

$$o^r = \sum_{i \in S^R} (s_i \in Q^{\neg R}) \tag{3.22}$$

$$o = C^v o^v + C^p o^p + C^u o^u + C^r o^r \tag{3.23}$$

Objective (3.19) calculates the total number of overstows. $ov(i)$ is the number of overstowing containers in a cell represented by its aft slot i. We have:

$$ov(i) = \begin{cases} 2 & \text{if } s_i \in Q^{20} \wedge p_i > minP(be(i)) \wedge p_{i+1} > minP(be(i)) \\ 1 & \text{if } (s_i \in Q^{40} \wedge p_i > minP(be(i))) \vee \\ & \quad (s_i \in Q^{20} \wedge (p_i > minP(be(i)) \oplus p_{i+1} > minP(be(i)))) \\ 0 & \text{otherwise} \end{cases}$$

where $be(i)$ is the set of slots below slot i in the same stack, $minP(S')$ is the earliest POD among the containers assigned to a set of slots S', and $\oplus$ denotes the *exclusive-or*

boolean operator. The empty stack objective (j), is represented by (3.20). The smallest POD index, 0, is assigned to virtual containers. Thus, when a stack i is empty, the sum of the values assigned to the subset of *Ports* variables in i is 0, otherwise the stack is being used. Objective (3.21) calculates the number of different POD of containers stowed in each stack, and objective (3.22) counts the number of non-reefer containers stowed in reefer slots. Objective (3.23) defines the cost function of the *CSPBDL*. The branch and bound algorithm applied to solve this problem constrains the cost variable o of the next solution to be lower than that of the solution with the lowest cost found so far. The constraints of the CP model are given by:

$$channeling(Conts, Slots) \tag{3.24}$$

$$c_{fore(i)} = c_i + 1, \qquad \forall i \in \{1, ..., |Q^{40A}|\} \tag{3.25}$$

$$element(s_i, l_i, L^C) \qquad \forall i \in S \tag{3.26}$$

$$element(s_i, h_i, H^C) \qquad \forall i \in S \tag{3.27}$$

$$element(s_i, w_i, W^C) \qquad \forall i \in S \tag{3.28}$$

$$element(s_i, p_i, P^C) \qquad \forall i \in S \tag{3.29}$$

$$s_{pos(j)} = j \qquad \forall j \in Q^L \tag{3.30}$$

$$regular(Lengths_i^{\pi}, R) \qquad \forall \pi \in \{A, F\}, i \in J \tag{3.31}$$

$$s_i \notin Q^{20R} \qquad \forall i \in S^{\neg R} \tag{3.32}$$

$$s_i \notin Q^{40R} \qquad \forall i \in S^{\neg RC} \tag{3.33}$$

$$s_i \in Q^{20} \qquad \forall i \in S^{20} \tag{3.34}$$

$$s_i \in Q^{40} \qquad \forall i \in S^{40} \tag{3.35}$$

$$\sum_{i \in S_j^{\pi}} h_i \leq hs_j \qquad \forall \pi \in \{A, F\}, i \in J \tag{3.36}$$

$$\sum_{i \in S_j} w_i \leq W_j^S \qquad \forall j \in J \tag{3.37}$$

Constraint (3.24) connects the two viewpoints such that both sets of variables *Conts* and *Slots* always have the same level of information. $fore(i)$ is the *Fore40* container bound to *Aft40* container i. Constraint (3.25) guarantees that the *Aft40* and *Fore40* part of a *40'* container are stowed in the same cell. Element constraints are used to bind all auxiliary variables introduced in the model to a viewpoint. Constraints (3.26) , (3.27), (3.28), and (3.29), bind each slot variable to the auxiliary variables representing the length, height, weight, and POD of the container stowed in the slot. *Release* containers are stowed in their pre-defined slots by constraint (3.30), where $pos(j)$ is the slot occupied by release container j. The valid patterns that containers stowed in stacks must follow according to their length are defined by (a) and (b). After assigning a length of 0 to virtual containers, we define a regular expression $R = 20^*40^*0^*$ that recognizes all the valid patterns according to these two constraints. Constraint (3.31) introduces a regular constraint for each aft and fore stack in order to restrict their stacking patterns

to follow those defined by R. Constraints (3.32) and (3.33) model the reefer constraint (c). Constraints (3.34) and (3.35) restrict the domains of slots that have just *20'* or *40'* container capacity to be the set of *20'* and *40'* containers, respectively. The height limit of each stack in the location is constrained by (3.36). All containers stowed in each side of a stack must be less or equal to the variable representing the height limit of the stack[4]. Constraint (3.37) restricts the weight of all containers stowed in a stack to be within the limits.

3.6.1 Symmetry-Breaking and Implied Constraints

We introduce a set of constraints to the CP model that aim at reducing the search space size and increase propagation. These constraints are implied by existing constraints and break symmetries either already present in the problem or introduced by our model representation.

$$exactly(\mathcal{V}, 0, |Q^V|) \qquad \forall\, \mathcal{V} \in \{Ports,\ Weights,\ Heights\} \tag{3.38}$$

$$exactly(Ports, p, Q^{P=p}) \qquad \forall\, p \in P^C \tag{3.39}$$

$$exactly(Weights, w, Q^{W=w}) \qquad \forall\, w \in W^C \tag{3.40}$$

$$exactly(Heights, H^\alpha, Q^\alpha) \qquad \forall\, \alpha \in \{DC, HC\} \tag{3.41}$$

$$h_i = h_k \qquad \forall\, j \in J,\ i,k \in \{(i,k)|\ i \in S_j^A, k \in S_j^F.eqCell(i,k)\} \tag{3.42}$$

$$sort(Conts^V) \tag{3.43}$$

$$s_i \notin Q^{40A} \qquad \forall\, i \in S^F \tag{3.44}$$

$$s_i \notin Q^{40F} \qquad \forall\, i \in S^A \tag{3.45}$$

$$s_i \leq s_k \qquad \forall\, j \in J,\ i,k \in \{(i,k)|\ i \in S_j^A, k \in S_j^F.eqType(i,k)\} \tag{3.46}$$

$$sort(Slots_j^E) \qquad \forall\, j \in J \tag{3.47}$$

$$lex(Class^i) \qquad \forall\, i \in Classes \tag{3.48}$$

Constraints (3.38), (3.39), (3.40), and (3.41) are implied constraints meant to improve the propagation power of the solver with respect to the auxiliary variables $Ports$, $Weights$, and $Heights$. Each individual auxiliary variable z_i is linked to a slot variable s_i with an element constraint. This ensures correctness but leads to weak propagation between the two sets of variables due to a lack of global perspective by the element constraints. To improve this, we first assign the value zero to the weight, height, and POD of virtual containers, (these containers are not supposed to affect total height, weight, or overstowage of each stack). Then, constraint (3.38) limits the number of

[4]The *HS* variables are not necessary to define the height constraint but play an important role in the height constraint lower bound introduced in Section 3.6.3.

variables set to zero from $Ports$, $Weights$, and $Heights$ to be the exact number of virtual containers. Additionally, constraints (3.39), (3.40), and (3.41) restrict the number of variables from $Ports$, $Weights$, and $Heights$ assigned to each possible POD, as well as weight or height to match the total number of containers with such features, respectively. Constraint (3.42) restricts the height of the containers stowed in the aft and fore slots of a cell to be equal. This is possible since both slots in a cell must be either empty or occupied at the same time (g), and there are no *20'* high-cube containers available to stow. The function $eqCell(j, k)$ indicates that two slots j and k belong to the same cell.

The weight of the containers make each of them almost unique, limiting the possibility of applying symmetry breaking constraints. It is possible, however, to break some of the symmetries introduced into the problem by our model representation. First, since all virtual containers have the same features, it is not relevant where each container is stowed. Constraint (3.43) posts a sorting constraint over the virtual containers, forcing the slots where these containers will be stowed to follow a non-decreasing order. Second, splitting *40'* containers into *Aft40* and *Fore40* parts also generates symmetrical solutions that are broken by constraint (3.44) and (3.45). Third, constraint (3.46) limits the possibility of swapping containers between two slots of a cell that have the same features. The function $eqType(j, k)$ indicates that two slots j and k belong to the same cell and have the same features, i.e., same reefer plug and length restrictions. Fourth, when all containers have the same POD, symmetrical solutions are generated by swapping containers stowed in slots with the same features within the same stack. Constraint (3.47) sorts in a non-decreasing order, the indices of the containers stowed in slots with the same features of each stack. Indices are assigned to containers such that conflicts between constraint (3.47) and valid stacking patterns (3.31) are avoided. *20'* containers are assigned a lower index than *40'* containers, and virtual containers have the highest index possible. Finally, symmetries between stacks with identical characteristics are considered. Stacks are classified according to their features: slot capacity, reefer capacity, height and weight limit. Constraint (3.48) removes symmetrical solutions generated by the containers stowed in similar stacks being swapped with each other, by requiring a lexicographical ordering on the indices of the containers stowed in these stacks.

3.6.2 Branching Strategies

Our branching strategy takes advantage of the structure of the model and uses the sets of different auxiliary variables in order to find high-quality solutions early in the search. We decompose the branching process into four sub-branchings: the first one focuses on finding high-quality solutions, the second and third on the feasibility the of two problematic constraints, and the fourth finds a valid assignment for the decision variables $Slots$. A detailed description of our branching strategy can be found in Delgado et al. (2010). In the case of the first sub-branching, since three of the four objectives of the *CSPBDL* rely on the POD of the containers, we start by branching over

the set of POD variables $Ports$. Variables bound to slots with containers that favor the clustering and overstowage objectives among the first free slots bottom-up of all stacks are preferred. After assigning all variables in $Ports$, we branch over the height and weight variables, $Heights$ and $Weights$. We start by branching over $Heights$ following a best-fit decreasing approach for selecting a stack, then we assign the smallest height possible among that of the containers to be stowed into the first free slot bottom-up in the stack. We use a similar approach for $Weights$, where the best fit is considered to be the stack with the greatest amount of free weight. Finally, we branch over $Slots$ in order to generate a concrete stowage plan. The domain size of variables in $Ports$ are considerably smaller than any of the viewpoints, making the process of finding valid assignments for $Ports$ easier. Once a valid slot plan is found, most of the time the search algorithm backtracks directly to the $Ports$ variables in order to find solutions with a lower cost. Therefore, a large part of the search process concentrates on a much smaller sub-problem. Branching is performed over the remaining variables only when a solution with a lower cost is likely to be found.

3.6.3 Lower Bounds

Five domain pruning rules are defined over partial solutions. Each rule solves a relaxed version of a sub-problem related to an objective or a constraint, generating lower bounds for their corresponding objectives and pruning values from the domain of the variables in the scope of the constraint. The lower bounds are described in more detail in Delgado et al. (2010).

Overstowage To calculate a lower bound on the overstowage of a partial solution ρ, we define a new function $min\overline{P}(S') = min_{i \in S'}(\overline{p}_i \,|\, p_i \in Ports)$ that selects the minimum upper bound $\overline{p}_i$ among the variables in $Ports$ corresponding to a subset S' of slots. Let $ov_\rho(i)$ be a function identical to $ov(i)$ where $min\overline{P}$ substitutes $minP$. It is easy to show that $ov_\rho(i)$ is a lower bound of $ov(i)$ for any completion of ρ (Delgado et al. (2010)). The pruning effect of the lower bound is achieved by adding the constraint $o^v \geq \sum_{i \in S^A} ov_\rho(i)$. An additional pruning rule can be applied when the domain of o^v has been reduced to a single value that is equal to the lower bound. In this situation, we enforce that all containers below non-overstowing containers are discharged at a later port:

$$|D(o^v)| = 1 \wedge \sum_{i \in S^A} ov_\rho(i) = o^v \rightarrow$$
$$\forall i \in \{k \in S^A \,|\, ov_\rho(k) = 0\}, j \in be(i) \,.\, p_i \leq p_j.$$

Empty stack In the remainder, we refer to container i as *unstowed* if the domain of c_i has more than one element. For the empty stack lower bound, a relaxation of the stowage problem is solved. The height capacity of the stacks is the only constraint considered and the containers to stow $Q^N_\rho = \{i \in Q \,|\, i \notin Q^V, |D(c_i)| > 1\}$

are accounted as normal height containers. We first consider the *used stacks* of ρ, $J^U_\rho = \{j \in J \,|\, \exists i \in S_j, |D(s_i)| = 1, s_i \notin Q^V\}$, where there are containers already stowed. The lower bound procedure stows as many containers as possible from Q^N_p in J^U_ρ such that the height capacity constraint is fulfilled. Once the used stacks are completely filled up, the empty stacks are sorted in decreasing order by height capacity and filled up with the remaining containers of Q^N_ρ. The minimum number of used stacks Min^u_ρ is the sum of used stacks $|J^U_\rho|$ plus the empty stacks necessary to stow all remaining containers. Min^u_ρ is a lower bound of the number of used stacks of any completion of ρ since the approach to solve the relaxed problem uses a minimum number of stacks. The pruning effect is achieved by adding $o^u \geq Min^u_\rho$.

Pure stack As with the used stack lower bound, a relaxed assignment problem is solved considering just the height capacity constraint and all containers not yet stowed as normal height containers. First, we introduce an alternative definition of the pure stack objective. Let $\Phi_i = |\{j \in J | \exists k \in S_j \,.\, p_k = i\}|$ be the number of stacks where at least one container with POD i is stowed. We can express the pure stack objective as $o^p = \sum_{i \in P} \Phi_i$. For this definition of the pure stack objective, we introduce a lower bound for a partial solution ρ. Let $Q^{N,P=i}_\rho$ be the set of unstowed containers in ρ with POD i, $J^{P=i}_\rho$ be the set of stacks stowing at least one container with POD i, and $J^{\neg P=i}_\rho = J \setminus J^{P=i}_\rho$ be the set of stacks where no container with POD i is stowed. Our goal is to generate a lower bound $Min^p_\rho(i)$ independently for each Φ_i, based on the approach followed to generate lower bounds for the used stacks objective. The pruning effect is achieved by adding $o^p \geq \sum_{i \in P} Min^p_\rho(i)$.

Reefer A lower bound Min^r_ρ for the reefer objective of a partial solution ρ can be deduced from a counting argument. Let $\Phi^{\neg R}_\rho = |\{i \in S^R \,|\, s_i \notin Q^R, \, |D(s_i)| = 1\}|$ denote the number of reefer slots stowing a non-reefer container in ρ. Clearly, $o^r \geq \Phi^{\neg R}_\rho$ for any completion of ρ. We tighten the lower bound of the reefer objective by considering the unstowed reefer containers and the reefer slots with more than one container in their domain that will not stow a virtual container. Let Ψ^R_ρ denote the number of unstowed reefer containers in ρ. Further, let Φ^{UR}_ρ be the number of reefer slots where no virtual container will be stowed. If $\Phi^{UR}_\rho > \Psi^R_\rho$ then at least $\Phi^{UR}_\rho - \Psi^R_\rho$ extra reefer slots will stow non-reefer containers. Thus, we can tighten Min^r_ρ as follows

$$Min^r_\rho = \begin{cases} \Phi^{UR}_\rho - \Psi^R_\rho + \Phi^{\neg R}_\rho & : \text{ if } \Phi^{UR}_\rho > \Psi^R_\rho \\ \Phi^{\neg R}_\rho & : \text{ otherwise.} \end{cases}$$

The pruning effect is achieved as usual by adding $o^r \geq Min^r_\rho$.

Height The domains of auxiliary variables from sequences *Height* and *HS* are tightened, and some conditions necessary for a partial solution to be viable are checked by solving three relaxed problems. First, the number of normal and high-cube containers that can possibly be stowed in the remaining free space of each stack is calculated. A

stack j of some partial solution ρ has free height $h_\rho(j) = \overline{hs}_j - h^s_j$, where h^s_j denotes the height of the stowed containers in stack j. Let $M^{DC}_\rho(j)$ and $M^{HC}_\rho(j)$ denote the maximum number of normal and high-cube containers that can be placed in stack j, respectively. We then have:

$$\begin{aligned} M^N_\rho(j) &= \lfloor h_\rho(j)/H^{DC} \rfloor, \\ M^{HC}_\rho(j) &= \lfloor h_\rho(j)/H^{HC} \rfloor. \end{aligned}$$

Let Ψ^{DC}_ρ and Ψ^{HC}_ρ denote the number of unassigned normal and high-cube containers of ρ, respectively. Then, all possible slot plans generated from partial solution ρ must satisfy

$$\sum_{j \in J} M^{DC}_\rho(j) \geq \Psi^N_\rho \quad \wedge \quad \sum_{j \in J} M^{HC}_\rho(j) \geq \Psi^{HC}_\rho.$$

Second, since containers cannot hang in the air, they must be stowed consecutively, bottom-up in all stacks. Therefore, when the sum of the height of containers stowed below tier n is equal to $\overline{hs}_j$, slots above tier n will not stow real containers. We stow virtual containers in slots of stack j that are above its height upper bound $\overline{hs}_j$. In the cases where the height of the container to be stowed is not yet known, it is assumed that the container will have normal height, since this generates an upper bound in the number of slots used in stack j. Additionally, the virtual containers are removed from slots that are below $\underline{hs}_j$, since these slots must stow real containers. Finally, we update $\underline{hs}_j$ by applying the bin packing propagation rule suggested in Shaw (2004)

$$\underline{hs}_j \geq \sum_{i \in Cont} H^c_i - \sum_{i \in J \setminus \{j\}} \overline{hs}_i, \quad \forall j \in S.$$

3.7 Experiments

We have defined two experimental setups to evaluate the models introduced in this chapter. In the first setup, we evaluate implementations of the CP and IP models in 236 slot planning instances, that have been derived from complete stowage plans provided by our industrial collaborator. Each instance is made by restowing a random location in one of the stowage plans. Since the plans have been applied in real life, we can assume that the containers have been assigned to locations according to the preferences of stowage coordinators. In the second setup, we incorporate the *CSPBDL* model that performs better in our first experiments, into the stowage planning module introduced in Pacino et al. (2011). We use this module to generate stowage plans for 18 industrial stowage conditions and measure the performance of our model.
The penalties for each objective term have been agreed with our industrial partner and have been set to 100 units for overstowing containers, C^v, 20 units for each POD in a stack, C^p, 10 units for using a new stack, C^u, and 5 units for stowing a non-reefer containers in a reefer slot, C^r.

3.7.1 Restowing Location Stowage Plans

In this experimental setup, the CP and IP models have been implemented in Gecode 3.3 (Gecode Team (2006)) and CPLEX 12.2, respectively. All the experiments were run on a Linux machine with two Quad Core Opteron processors at 1.7 GHz and 8 GB of memory. In order to characterize hard instances, we have partitioned them into groups with different features. Table 3.3 presents the features of each group of instances. Notice that the instances have a low number of POD. This is to expect from high quality stowage plans that avoid POD mixing. This does not imply that the instances are easy since instances with a single POD still embeds the bin-packing problem.

Grp.	#inst.	Cap. (TEUs)	Cont. (TEUs)	*40'*	*20'*	R	HC	#POD		
		min , max , avg	min , max , avg					1	2	≥ 3
1	13	16 , 116 , 63	8 , 116 , 54	*				13		
2	22	8 , 168 , 68	8 , 136 , 52		*			22		
3	13	30 , 124 , 74	8 , 124 , 68	*	*			13		
4	78	6 , 208 , 79	2 , 202 , 63	*			*	78		
5	36	38 , 176 , 97	8 , 170 , 81	*	*		*	36		
6	15	42 , 172 , 73	16 , 74 , 46	*		*		15		
7	14	72 , 204 , 147	24 , 202 , 117	*	*	*	*	14		
8	14	40 , 148 , 96	40 , 136 , 87	*		*	*		14	
9	17	44 , 220 , 124	36 , 200 , 111	*	*	*	*		15	2
10	8	72 , 176 , 122	10 , 156 , 93	*		*	*		6	2
11	6	48 , 176 , 101	28 , 148 , 84	*	*	*	*		3	3

Table 3.3: *Grouping of instances.* The first column is the group id and the second number of instances in the group. The minimum, maximum, and average capacity of locations (Cap.), and number of containers to stow (Cont.), in TEUs are presented in the next six columns. The remaining columns show the features of the containers present in each group (*40'*, *20'*, reefer, and high-cube), and the number of instances with 1, 2 or 3 different POD in the group.

Figure 3.2 shows the space utilization as a function of location size. The space utilization is the number of TEUs to stow divided by the total TEU capacity of the location. 172 of the 236 instances have a space utilization above 80%, which is to expect from real stowage plans.

Impact of CP enhancements

We first analyze the impact of the different enhancements of the CP model introduced in Section 3.6. We define four CP models. The *basic* model includes only the core constraints and objectives of the *CSPBDL* (3.19 - 3.37). A simple branching strategy is used in this model, where the stacks are filled bottom-up from left to right, and the container with the smallest index in the domain of the slot variable to be branched on, is stowed in the slot. The *improved* model includes the symmetry-breaking and implied

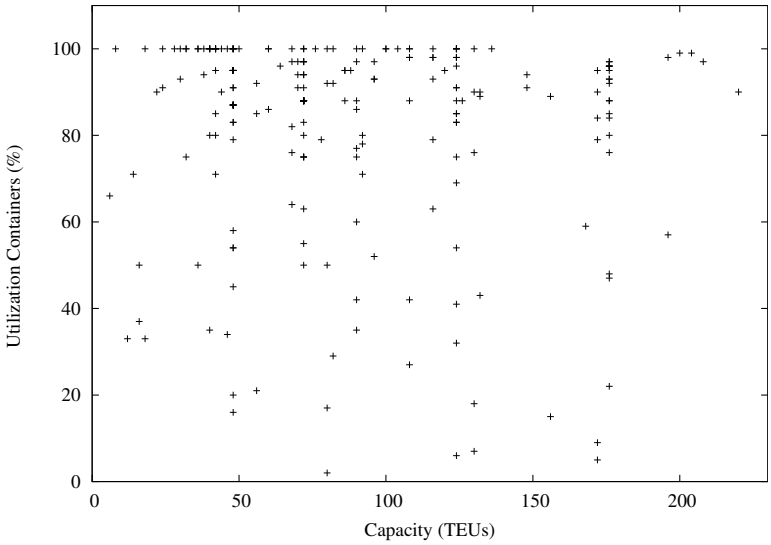


Figure 3.2: Utilization as a function of location capacity.

constraints from Section 3.6.1. Its branching strategy is similar to the *basic* model, however, the containers are assigned indices based on their features to avoid conflicts with some of the new constraints introduced. Finally, the *branching* and *advanced* models include the tailor-made branching strategy introduced in Section 3.6.2 and the lower bounds introduced in Section 3.6.3, respectively.

Recall that we aim at solving *CSPBDL* instances within one second. The solver can return an optimal solution before that, but after one second it must return its current solution. The results are summarised in table 3.4, where each row corresponds to one of the instance groups introduced in table 3.3.

Grp.	All (%)	**Basic**		**Improved**		**Branching**		**Advanced**	
		sol (%)	opt (%)	sol (%)	opt (%)	sol (%)	opt (%)	sol (%)	opt (%)
1	100	100	69	100	85	100	100	100	100
2	82	86	54	82	50	91	90	91	91
3	92	92	61	92	69	100	100	100	100
4	74	82	11	83	61	96	90	96	95
5	36	53	11	69	44	89	75	92	89
6	80	80	7	93	40	100	93	100	93
7	36	71	0	64	14	64	29	64	57
8	50	79	14	79	29	93	93	93	93
9	47	76	18	65	12	76	47	76	76
10	75	62	12	87	37	100	62	100	100
11	67	67	17	83	33	83	67	83	83
Total	66	77	21	80	48	89	80	92	90

Table 3.4: *Percentage of instances solved and proven to optimality by the CP models.* The first column provides the id of the group, the second one shows the percentage of instances of the group solved by all CP models, and the subsequent columns the percentage of instances solved and proven optimal by each model independently. The last row presents the results for the complete set of instances.

The total number of instances solved and proven optimal increases for each extension of the basic model. A careful inspection of the table shows that this does not apply to all groups individually, but overall the impact of the model improvements are quite similar for each group. The total time needed for processing all instances is also reduced considerably for each model, from 186.3 seconds for the basic model, to 126.9 seconds for the improved model, 56.8 seconds for the branching model, and 34 seconds for the advanced model. Instances not solved were accounted for by one second in the total time.

To compare the runtimes of individual instances, we analysed the subset of 156 instances (66.1%), solved by all four CP models, (second column of Table 3.4). The plot in Figure 3.3, shows the runtime of the models for each instance, where label BA represents the basic model, IM the improved model, and BR and AD the branching and advance models, respectively. We have sorted the instances such that the expected runtime dominance between the models is clearly observable.

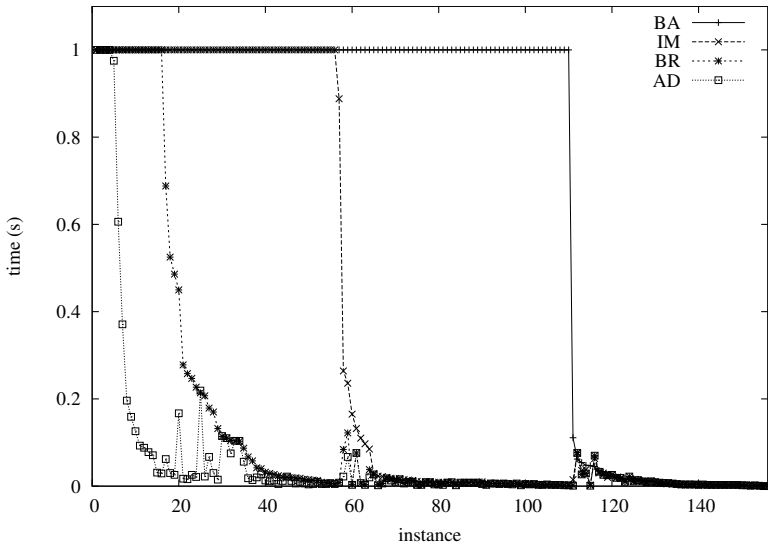


Figure 3.3: Runtime of the four CP models.

The plot in Figure 3.4 shows in the optimality gap of 39 out of the 156 instances, that at least one model solved sub-optimally. The labels follow the same conventions as in the left graph. Again, we have sorted the instances to highlight a quite robust optimality dominance between the models.

Wrt. the hardness of different instance groups, the results in Table 3.4 indicate that instances with many features (group 8-11), are slightly harder than instances with few features. In particular, instances only stowing *40'* containers (group 1 and all even groups except 2), are significantly easier than instances stowing both *20'* and *40'* containers. Interestingly, there is no significant positive correlation between space utilization and runtime. This is surprising as one would expect the hardness of slot planning problems to increase with space utilization.

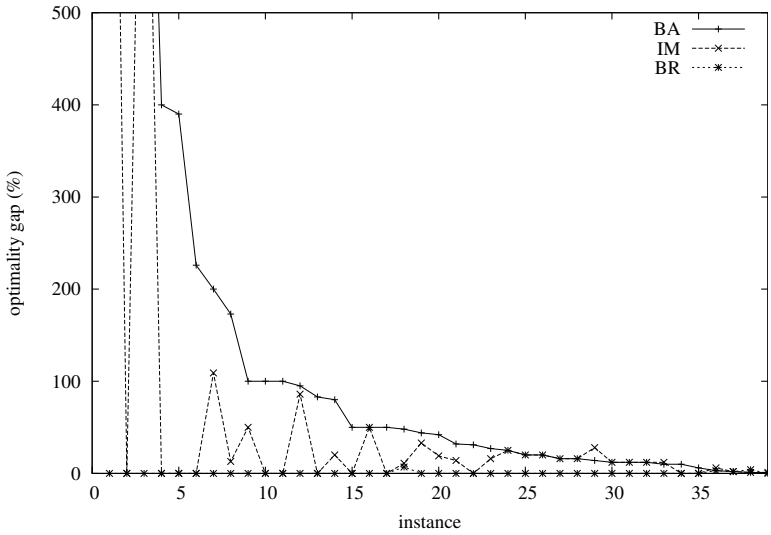


Figure 3.4: Optimality dominance of the three CP models with gaps.

Comparing the performance of the IP and CP models

In this section, we compare the performance of the IP and CP model using a one second and 10 seconds runtime limit. Table 3.5 summarizes the percentage of instances solved, proven to optimality, and runtime of the models over the instance groups. The runtime of each instance group is calculated by summing up the time taken for solving all the instances in the group. Unsolved instances were accounted for by the time limit of the experiment. The upper table shows the results of the experiments with a one second time limit. The CP model is clearly dominant in all groups in both, number of instances solved and instances solved optimally. In total, the CP model solves 28% more instances than the IP model in just a quarter of the time. When the time limit is extended to 10 seconds (lower table in Table 3.5), the IP model increases the number of instances solved by a 22% and the ones solved optimally by 17%. The runtime, however, increases by a factor of five. The instances solved by the CP model increase by a modest 1.7%, and the ones solved optimally by 3.8%, while the runtime increases a factor of six. The total time of the CP model, however, is still around a quarter of the IP model.

The hardness of instances for the IP model is similar to that of the CP model: instances with many features are slightly harder than instances with few, and instances stowing *40*' containers only are easier than instances stowing both *20*' and *40*' containers.

As in the CP experiment, we now focus on the instances solved by both models. Table 3.6 summarizes the results of the experiments with 1 and 10 seconds time limit, showing the percentage of instances solved by both models, only the IP model, only the CP model, and none of them. For the one second time limit experiment, a total of 148 instances (63%), are solved by both models. All solutions produced by the CP model

Grp.	CP 1 sec.			IP 1 sec.		
	sol (%)	opt (%)	time (s)	sol (%)	opt (%)	time (s)
1	100	100	0.1	100	100	1.7
2	**91**	**91**	3.6	59	59	11.6
3	**100**	**100**	0.5	54	54	8.9
4	**96**	**95**	6.0	87	79	27.7
5	**92**	**89**	7.1	33	28	31.3
6	100	**93**	1.2	100	93	4.0
7	**64**	**57**	6.8	29	21	11.2
8	**93**	**93**	1.5	43	36	10.6
9	**76**	**76**	5.2	24	24	14.8
10	**100**	**100**	0.7	62	62	4.4
11	**83**	**83**	1.3	50	50	3.5
Total	**92**	**90**	34	64	59	129.8

Grp.	CP 10 sec.			IP 10 sec.		
	sol (%)	opt (%)	time (s)	sol (%)	opt (%)	time (s)
1	100	100	0.1	100	100	1.8
2	91	91	21.6	**95**	91	50.4
3	**100**	**100**	0.5	92	85	35.3
4	**99**	**99**	19.7	96	94	87.0
5	**92**	**92**	39.0	72	56	192.0
6	100	**100**	5.4	100	93	13.0
7	64	**64**	53.5	64	29	102.8
8	**93**	**93**	10.5	79	64	74.1
9	**88**	**88**	36.5	53	41	112.3
10	**100**	**100**	0.7	88	62	31.5
11	**83**	**83**	10.3	67	50	30.5
Total	**94**	**94**	198	86	76	730.9

Table 3.5: *Results for the CP and IP models with a 1 second (upper table), and 10 seconds (lower table), limit.*

are optimal. The IP model produced 2 suboptimal solutions with an optimality gap of 67% and 214%. For the 10 seconds time limit experiment, the number of instances solved by both models increased by 19% (45 instances). The number of suboptimal solutions also increased for the IP model (from 2 to 8), with the optimality gap ranging from 3.1% to 317%. The suboptimal instances are spread over 6 different groups.

Grp.	1 sec.				10 sec.			
	both (%)	IP (%)	CP (%)	none (%)	both (%)	IP (%)	CP (%)	none (%)
1	100	0	0	0	100	0	0	0
2	55	5	35	5	82	14	4	0
3	54	0	46	0	92	0	8	0
4	86	1	10	3	95	1	4	0
5	33	0	58	9	67	6	24	3
6	100	0	0	0	100	0	0	0
7	29	0	36	35	50	14	22	14
8	43	0	50	7	71	8	21	0
9	24	0	53	24	53	0	35	12
10	62	0	38	0	88	0	12	0
11	50	0	33	17	67	0	17	16
Tot	63	1	29	7	82	4	11	3

Table 3.6: *Results for the instances solved by any of the two models with 1 and 10 seconds time limit.*

3.7.2 Slot Planning Master Plans

Given that the CP model outperformed the IP model in the previous experiments, we move a step forward and incorporate our CP model into a module for the automatic generation of stowage plans introduced in Pacino et al. (2011). The CP model becomes part of the slot planning component of this module, stowing the containers assigned to each location independently. We limit slot planning to the current port, since this is the only port that stowage coordinators stow. In the cases where the CP model cannot stow the containers assigned to a location (8% in the previous experiments), a modified version of the LS algorithm from Pacino and Jensen (2012), that allows rolling out containers is used to generate the location slot plan. Removing a few containers from the vessel has little impact on its stability.

For this experiment, we aim at generating stowage plans for 18 industrial stowage conditions. In Pacino et al. (2011), the master planning phase problem is formulated as a mathematical programming model. We solve this mathematical model as a MIP and an IP model, and incorporate an evaluated impact of this decision into the generated stowage plans. Master plans with a 5% optimality gap and a 3,600s time limit are acceptable for slot planning. Solving the MIP model is considerably faster than the IP model, but it has some drawbacks. The master planning IP model assigns integer numbers of containers to locations. The MIP model, however, may assign fractional numbers of containers to be stowed in locations, something that is physically impossible. To deal with this issue, we attempt to stow a unit in the first location with available capacity, where the largest container fraction has been assigned by the master plan. If none of the locations have capacity left, the container is rolled out. The quality of each slot plan is evaluated by comparing it with the best slot plan generated by CP for the same location within twenty minutes. In the cases where containers are rolled out by the LS algorithm, the slot plan is evaluated against the best plan generated by CP within twenty minutes, stowing the same containers as LS, that is, discarding the rolled out containers. The time limit to stow a location is one second. All experiments were run on a Linux machine with two Six Core AMD Opteron processors at 2.0 Ghz and 32 GB of memory, with the CP model being implemented using the library Gecode 3.5. Due to the non-deterministic nature of the LS algorithm, results of slot planning are reported on average over 10 runs of the algorithm. Table 3.7 summarizes the results of our experiments.

Vessels are slot planned fast by our approach. For the instances where the master planning phase was solved, slot plans are generated within 17 seconds in total. When the slot planning IP master plans (results of the master planning phase when solving it as an IP model), a reduction in the number of containers rolled out due to fractionality and by odd numbers of *20'* containers (remember that a cell cannot stow a single *20'*), in locations is observed in most of the instances. This is, however, a very small (0.4% max), fraction. The maximum roll-out of an instance is 5.99%, a reasonable number given the number of containers typically rolled from a loadlist by stowage coordinators. Only two instances have an average optimality gap over 10%, due to the presence of

Slot Planning Current Port											
ID	Conts.	Time(s)		Locs.		Frac.+Odd		Rolled out(%)		Gap(%)	
		Cont.	Int.	Cont.	Int.	Cont.	Int.	Cont.	Int.	Cont.	Int.
1	990	3.48	4.96	34	34	4	2	0.40	0.20	0.00	0.00
2	2222	13.21	13.10	61	57	15	6	2.58	2.04	3.05	4.60
3	783	4.07	4.94	33	32	3	7	0.38	0.89	0.36	0.00
4	1876	16.44	13.64	69	70	14	5	1.76	1.08	12.77	10.47
5	573	8.94	6.08	29	25	10	5	1.75	0.87	1.41	0.45
6	862	6.83	5.26	27	35	5	6	0.93	0.70	1.25	0.00
7	238	1.52	4.08	12	14	1	1	0.42	0.42	0.00	0.00
8	334	0.99	0.66	14	13	4	4	2.10	1.20	1.57	0.00
9	1064	9.20	11.06	29	32	8	5	4.68	5.99	13.86	16.44
10	314	0.95	-	12	-	1	-	0.32	-	0.00	-
11	757	6.97	7.96	37	27	7	5	0.92	0.66	0.63	0.59
12	894	9.69	9.19	24	23	0	2	0.00	0.22	0.44	0.67
13	2190	13.57	10.73	61	58	7	5	1.30	0.23	0.78	0.03
14	901	5.21	-	30	-	14	-	2.51	-	6.73	-
15	302	1.79	1.46	16	14	2	6	0.66	1.99	0.00	0.00
16	1618	9.73	13.98	50	50	4	6	0.61	2.94	0.38	1.54
17	200	0.94	0.91	12	13	0	0	0.00	0.00	0.00	0.00
18	1261	9,82	7,73	38	37	6	4	0.48	0.40	2,01	9,75

Table 3.7: *Slot Planning MIP (Cont.) and IP (Int.) master plans with a 5% gap.* The first and second columns are the id of the instance and the number of containers to stow in the first port. The next columns show grouped results of slot planning, based on MIP and IP master plans. The third and fourth columns depict the runtime for the slot plans, the fifth and sixth columns are the number of locations. The seventh and eighth columns total the number of rolled out containers by fractionality and by an odd number of *20'* types, and finally the ninth and tenth columns are the percentage of total containers rolled out. The last two columns show the average gap of the slot plans. A dash indicates that no master plan was provided.

outliers, and the median of all instances is always 0%. CP generated optimal slot plans, within one second for 91.8% of the locations of the MIP master plans and 94.8% of the locations of the IP ones. Moreover, CP was able to prove optimality in 83% of the slot plans generated for the locations from both, the MIP and IP master plans.

3.8 Conclusions

In this chapter, we have introduced an accurate model called *CSPBDL* for stowing a set of containers in a location. The *CSPBDL* is NP-hard and is an important sub-problem of the successful multi-phase approaches to stowage planning optimization. We have developed two CP and IP models to solve the *CSPBDL* optimally. Our experiments show that both models can be solved fast and that it is possible to improve the performance of the CP model, such that it can produce optimal solutions for 90% of the 236 industrial instances in less than one second, which is well within the time requirements for practical stowage support tools. We also tested the CP model, the best of the two models according to our experimental evaluation, as a component of a module for the automatic generation of stowage plans. In this setup, the CP model behaved similarly

to the first experiment, generating optimal slot plans for more than 90% of the locations and proving optimality for 83% of them. Future research includes improving the performance and stability of our solvers (e.g., diving heuristics and other techniques may be used to improve the IP model), and extending the *CSPBDL* to include on deck locations and special containers such as out-of-gauge, pallet-wide, and containers with dangerous goods.

Chapter 4

Accurate Master Planning

In this chapter, we present an accurate mathematical model for solving the first phase, the master planning phase, of our 2-phase decomposition approach for scalable stowage planning. During master planning, containers are assigned to vessel locations rather than individual slots. The mathematical model formulated in this chapter includes accurate stability and stress force calculations and ballast tanks. Ballast tanks are a feature of stowage planning, widely used by stowage coordinators, that have been disregarded in most of the previous academic work. This chapter presents an approach to address small variations in the vessel displacement, which is a consequence of including ballast tanks into the mathematical model that affects stability and stress calculations. This chapter is based on Pacino, Delgado, Jensen, and Bebbington (2012).

4.1 Introduction

All of the mathematical models present in the literature that consider accurate stability and stress forces are based on the assumption that the displacement of the ship (the total weight of the loaded vessel), is constant. Hydrostatics, such as buoyancy, stability, trim, and draft are non-linear functions of the ship's center of gravity and vessel displacement. They can, however, easily be linearized when the displacement is constant and translated to bounds on the position of the center of gravity, which is also simple to compute when the displacement is constant. Container vessels have ballast tanks that are used by stowage coordinators to better handle the stability of the vessel and apply stowage configurations that are otherwise unfeasible. These tanks can account for approximately 25% of the ship's total displacement and including them in a mathematical model brings forth a number of non-linear relations.

In this chapter, we present a mathematical model for the master planning phase that considers ballast tanks and deals with the variable displacement. This model is concerned only with the seaworthiness of the vessel, but can easily be extended to optimize other aspects of the plan such as crane utilization. According to our industrial collaborator, it is possible for stowage coordinators to make an educated guess

on the amount of ballast water that a vessel might need that is within 15% of the current amount in the tanks. We use this assumption to define a displacement range within which we are able to formulate a linearization of the stability constraints with an acceptable error. We analyse the accuracy of our model experimentally on 10 real instances provided by our industrial partner. Our analysis suggests that within 5% of the current displacement of the vessel the model is accurate enough to guarantee the seaworthiness of the vessel.

The remainder of the chapter is organized as follows. Section 4.2 introduces related work. Section 4.3 presents our model and sections 4.4 and 4.5 present the experimental analysis and conclusions.

4.2 Literature Review

Even though several of the publications on stowage planning available in the past years address stability and stress forces, very few present models that incorporate them as constraints or objectives. Additionally, none of them consider variable displacement due to ballast tanks. One of the most complete formulations of stability and stress moments as part of an IP model is introduced in Botter and Brinati (1992). Though their model is too complex to solve in practice, it constrains GM, transversal stability (heel angle), trim, shear forces, and bending moments. Linearizations depending on the displacement of the vessel (which is variable due to the inclusion of the loading and unloading sequence of containers into the model), are suggested for the stability constraints. No evaluation of the impact of these linearizations is presented, probably due to the fact that their model was not used in practice. Shear forces and bending moments are addressed, but they disregard the impact of the cargo on the buoyancy force. The IP formulations introduced in Ambrosino et al. (2004) and Li et al. (2008), assign containers from a loadlist to individual slots on the vessel. These models handle transversal stability and trim by constraining the weight difference between transversal (left and right), and horizontal (bow and stern), sections of the vessel to be within certain tolerance. The GM is constrained by not allowing heavy containers on top of light ones, a rule of thumb used in the industry for some vessels, but not necessarily for all vessels. In Kang and Kim (2002), a model that assigns types of containers to sections of the vessel is introduced. This model constrains the center of gravity of the vessel with respect to precomputed constants to satisfy GM, trim, and transversal stability constraints.

To the best of our knowledge, the only approach available in the literature that considers the use of ballast tanks is presented in Aslidis (1984). A heuristic uses ballast water to bring the longitudinal center of gravity within a permissible range defined, based on the trim desired. Later, a local search is used to fix the GM.

4.3 Stability and Stress Model with Ballast Tanks

The introduction of ballast tanks into the optimization model causes the displacement of the vessel to become variable. This makes the calculation of the center of gravity non-linear. Thus, it is no longer possible to use the linearization of the hydrostatic data introduced in Pacino et al. (2011). Two major non-linear relations must be handled once variable displacement has to be modelled. The first is the calculation of the center of gravity and the second is the linearization of the hydrostatic data. Consider the calculation of the longitudinal center of gravity (lcg) without ballast tanks:

$$\frac{LM^o + \sum_{l \in \mathcal{L}} D_l^L v_l}{W}, \tag{4.1}$$

where LM^o is the constant longitudinal moment of the vessel, D_l^L and v_l are the lcg and weight of a location $l \in \mathcal{L}$, respectively. The displacement given by $W = W^o + \sum_{l \in \mathcal{L}} v_l$, where W^o is the constant weight of the vessel. Since all containers in the loadlist are loaded, W is constant, which makes the calculation linear. Now consider the same calculation where we include ballast tanks as a variable:

$$\frac{LM^o + \sum_{l \in \mathcal{L}} D_l^L v_l + \sum_{u \in U} D_u^L v_u}{W + \sum_{u \in U} v_u}, \tag{4.2}$$

where D_u^L is the lcg of tank u, and v_u is the variable defining the amount of water to be loaded in tank $u \in U$. Since the amount of water in the tanks is not known a priori, the displacement of the vessel is no longer constant, and the calculation becomes non-linear. In order to deal with this, we propose the following approximation:

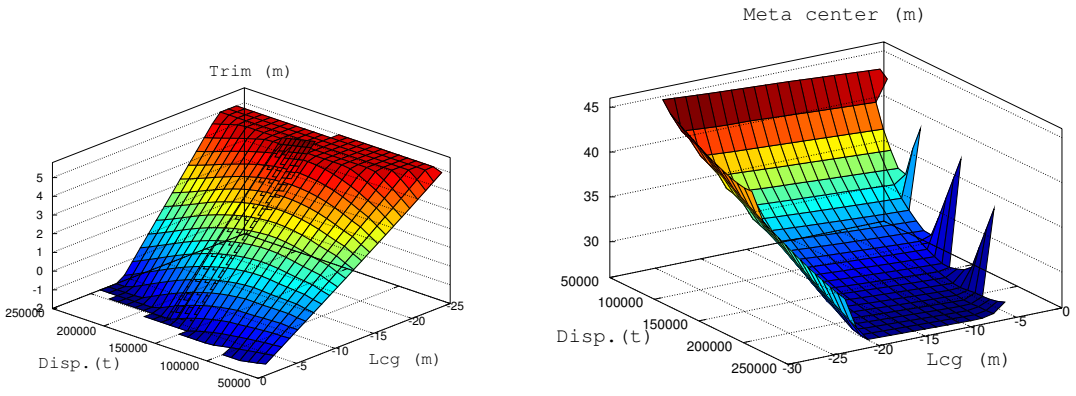
$$LCG = \frac{LM^o + \sum_{l \in \mathcal{L}} D_l^L v_l + \sum_{u \in U} D_u^L (v_u + \Delta_u)}{W + W^T + \sum_{u \in U} \Delta_u} \tag{4.3}$$

$$\approx \frac{LM^o + \sum_{l \in \mathcal{L}} D_l^L v_l + \sum_{u \in U} D_u^L (v_u + \Delta_u)}{W + W^T}, \tag{4.4}$$

where we model the ballast water estimation error of the stowage coordinator with the variables Δ_u. Thus, the total displacement becomes $W + W^T + \sum_{u \in U} \Delta_u$, where W^T represents the weight of the estimated ballast water. We then make a linear approximation of the lcg of the vessel by removing the allowed changes of ballast water from the denominator of the fraction, resulting in equation (4.4). Given the total capacity of the tanks (W^{BT}), the constant weight of an empty vessel $W^o \approx 2W^{BT}$, and the weight of the cargo $W^C \approx 6W^{BT}$, we can reasonably assume that the error in the approximation of the lcg (given that stowage coordinators can estimate the ballast within 15 % accuracy), is less than $0.15W^{BT}/(2W^{BT} + 6W^{BT} + W^{BT}) = 1.7\%$. Note that the same approximation can be used to calculate the vertical and transversal center of gravity.

The assumption that the weight of ballast water lies within a given interval is useful for the linearization of the hydrostatic calculations. Hydrostatic calculations are in practice linear approximations of given data points. When the center of gravity and the displacement of the vessel are known, the linearization is accurate. For the problem we are going to model, however, this is not the case since both the center of gravity and the displacement can vary. Figure 4.1 shows a plot of the hydrostatic data for the trim and metacenter calculation. The functions are clearly non-linear. Notice that within a small displacement interval it is possible, however, to approximate the functions accurately with a plane. This is only true for displacement levels that are not at the extremes of the data tables, but it is reasonable to assume that the displacement of a stowage plan will be within these extremes. The planes described above can be defined by the limited ballast water change and thus be used in our model.

The buoyancy of a vessel is the volume of water that the vessel displaces. In order



(a) (b)

Figure 4.1: (a) Trim as a function of displacement and lcg (b) Metacenter as a function of displacement and lcg.

to calculate this volume, it is necessary to know the shape of the vessel hull. For this purpose, the hydrostatic data tables provide the possibility of calculating the submerged area of a vessel at a specific point called a *station*. Figure 4.2 shows an example of such areas and how stations are distributed along the vessel. Given two adjacent stations, s_1 and s_2, the buoyancy of the vessel section between the two stations is approximated by $\frac{(A_{s_1}+A_{s_2})D(s_1,s_2)\delta^W}{2}$, where A_s is the underwater area at station s, from now on referred to as *bonjean*, $D(s_1, s_2)$ is the distance between the two stations, and δ^W is the density of the water. As Figure 4.2 shows, stations are not evenly distributed along the vessel. A greater concentration is found at the vessel's extremities, where the hull shape changes the most. Figure 4.3 shows two plots of the hydrostatic data related to the bonjean of a station at the bow and a station in the middle of a vessel. As expected, the function describing the hull at the bow is highly non-linear since the hull greatly changes, which is not the case for stations in the middle of the vessel. Within specific displacement

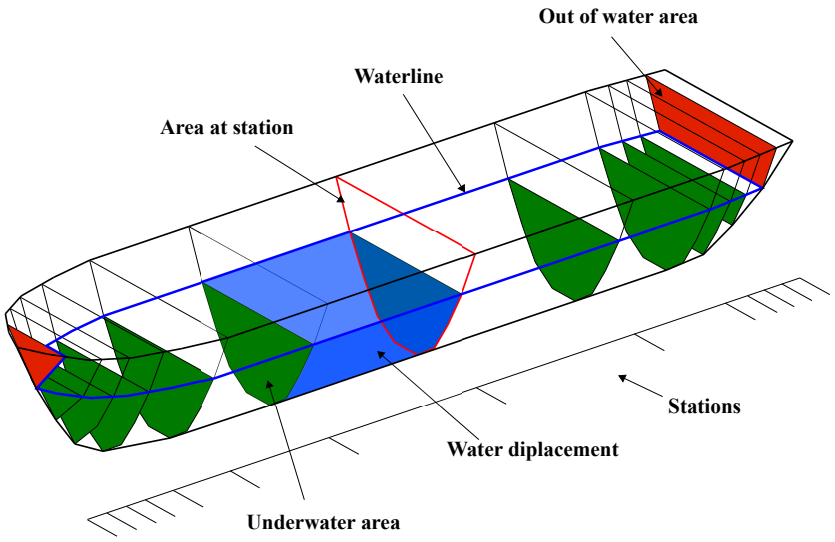
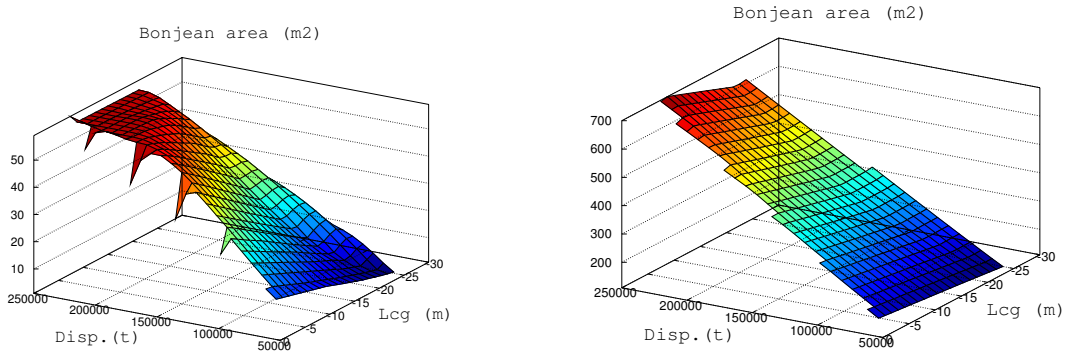


Figure 4.2: Areas for buoyancy calculation and distribution of stations

ranges, it is still possible to approximate the function linearly. Should one want to model displacement ranges that include the most non-linear parts, piece-wise linear approximations with a few binary variables can be used.



(a) (b)

Figure 4.3: (a) Underwater area as a function of displacement and lcg at a bow station (b) Underwater area as a function of displacement and lcg at a middle station.

4.3.1 A MIP Model

Following the linear approximation described in the previous section, we propose a MIP model for the master planning phase that includes ballast tanks modeling. Objectives that focus on efficiency of the master plan (described in Pacino et al. (2011)) are not included in the model under analysis as they do not have any influence on the

seaworthiness of the vessel and are thus irrelevant to this study. Containers are grouped into mutually exclusive types based on their features. A *container type* $\tau \in \mathcal{T}$ is a 4-tuple (l,h,r,w), where $l \in L$ is the length of the container in feet, $h \in H$ is the height of the container, $r \in R$ is the reefer properties of the container (reefer or non-reefer), and $w \in W$ is the weight class of the container. For the model presented in this chapter we define $\mathcal{T} = \{(l, h, r, w) \,|\, l \in L = \{20, 40\}, h \in H = \{DC\}, r \in R = \{RF, NR\}, w \in W = \{LW, HW\}\}$, where DC refers to dry-cargo containers, RF and NR to reefer and non-reefer containers, and LW and HW to light and heavy weight classes. Further, let $\mathcal{T}^{\alpha}$ be a subset of these types with an attribute value restricted to be α. For instance, $\mathcal{T}^{RF}$ represents the set of types with reefer capabilities.

When doing master planning it is necessary to consider the current port and the downstream ports to generate robust stowage plans. We define a *transport* as a pair of loading and discharging ports. Thus, the multi-port master planning phase assigns types of containers to locations during transports. Notice that the container types $\mathcal{T}$ are only a classification and thus are not bound to fixed weight ranges. For each type $\tau \in T$ the average weight of the containers during a transport is calculated, causing a more refined weight categorization than taking the average weight of the containers in each container class. We remind the reader that the sets and constants used in the model presented in this chapter have been previously defined in the section *Sets and Constants*, at the beginning of this document. Now we define the variables of the multi-port master planning model.

Decision variables

$x_l^{t\tau} \in \mathbb{Z}$	Number of containers of type t to be stowed in location l during transport t.
$x_{pu} \in \mathbb{R}^+$	Amount of ballast water in tons loaded at port p in tank u.

Auxiliary variables

$y_{pu} \in \mathbb{R}^+$	Change in ballast water at port p in tank u.
$v_p^W \in \mathbb{R}^+$	Vessel displacement.
$v_{pl}^W \in \mathbb{R}^+$	Weight of containers stowed at port p in location l.
$v_{pl}^{W\{20,40\}} \in \mathbb{R}^+$	Weight of *20'*, *40'* containers stowed in location l at port p.
$v_p^L \in \mathbb{R}$	Longitudinal center of gravity at port p.
$v_p^V \in \mathbb{R}^+$	Vertical center of gravity at port p.
$v_p^M \in \mathbb{R}^+$	Metacenter at port p.
$v_{ps}^B \in \mathbb{R}^+$	Buoyancy force at port p of section between stations s and $s+1$.
$v_{pf}^{S\alpha} \in \mathbb{R}^+$	Shear force at port p fore or aft of frame f.
$v_{pf}^{B\alpha} \in \mathbb{R}^+$	Bending moment at port p fore or aft of frame f.

The objective function of the MIP model for the evaluation of the accuracy of stability and stress force calculations when modeling ballast tanks is:

$$\textbf{minimize} \quad \sum_{p \in P} \sum_{u \in U} y_{pu} \tag{4.5}$$

The accuracy of the approximations decreases with the extent of the change in ballast water, thus, objective (4.5) minimizes this change. Now we introduce the capacity

constraints.

s.t.

$$\sum_{t \in TR_p^{ON}} \sum_{\tau \in \mathcal{T}} \Pi_\tau^{TEU} x_l^{t\tau} \leq K_{pl}^{TEU} \quad \forall p \in P, l \in \mathcal{L} \tag{4.6}$$

$$\sum_{t \in TR_p^{ON}} \sum_{\tau \in \mathcal{T}^\alpha} x_l^{t\tau} \leq K_{pl}^{\alpha} \quad \forall p \in P, l \in \mathcal{L}, \alpha \in L \tag{4.7}$$

$$\sum_{t \in TR_p^{ON}} \sum_{\tau \in \mathcal{T}^{RF}} x_l^{t\tau} \leq K_{pl}^{RF} \quad \forall p \in P, l \in \mathcal{L} \tag{4.8}$$

$$\sum_{t \in TR_p^{ON}} \sum_{\tau \in \mathcal{T}^{RF}} \Pi_\tau^{FEU} x_l^{t\tau} \leq K_{pl}^{RC} \quad \forall p \in P, l \in \mathcal{L} \tag{4.9}$$

$$\sum_{t \in TR_p^{ON}} \sum_{\tau \in T^\alpha} W_{t\tau} x_l^{t\tau} = v_{pl}^{W\alpha} \quad \forall p \in P, l \in \mathcal{L}, \alpha \in L \tag{4.10}$$

$$v_{pl}^{W20} \leq W_{pl}^{20} \quad \forall p \in P, l \in \mathcal{L} \tag{4.11}$$

$$0.5 v_{pl}^{W20} + v_{pl}^{W40} \leq W_{pl}^{40} \quad \forall p \in P, l \in \mathcal{L} \tag{4.12}$$

$$v_{pl}^{W20} + v_{pl}^{W40} = v_{pl}^{W} \quad \forall p \in P, l \in \mathcal{L} \tag{4.13}$$

All the weight limits and capacities of the model have been reduced to account for the release. The TEU capacity of each location is enforced by constraint (4.6). Location specific capacity requirements regarding the length of the containers are enforced using constraint (4.7). Constraint (4.8), and (4.9) limit, respectively the total number of reefer TEU and Forty-foot Equivalent Units (FEU) used. With constraint (4.10) we define the variables v_l^{W20} and v_l^{W40}, and use them in constraint (4.11) and (4.12) to limit the weight of the *20*' and *40*' containers stowed in a location. Constraint (4.13) defines the auxiliary variable v_l^W. Now we introduce the constraints, in order to load all containers and to limit the changes in ballast water.

$$\sum_{l \in \mathcal{L}} x_l^{t\tau} = LD_t^\tau \quad \forall \tau \in \mathcal{T}, t \in TR \tag{4.14}$$

$$x_{pu} \leq K_u \quad \forall u \in U \tag{4.15}$$

$$(E_u - \epsilon_u) \leq x_{pu} \leq (E_u + \epsilon_u) \quad \forall p \in P, u \in U \tag{4.16}$$

Constraint (4.14) ensures that all containers are loaded. Constraint (4.15) defines the capacity of the tanks and constraint (4.16) limits their change in ballast water. Rather than considering the allowed ballast change for all the tanks combined, constraint (4.16) limits the ballast change for each tank u independently. Though limiting the change of each ballast tank independently reduces the stowage configurations that can be applied, this approach minimizes the error in the calculations of the center of gravity of the tanks. Now we introduce the stability constraints of the model:

$$\sum_{u \in U} x_{pu} + \sum_{l \in \mathcal{L}} v_{pl}^W + W_p^O = v_p^W \quad \forall p \in P \tag{4.17}$$

$$\frac{\sum_{l\in\mathcal{L}} D_l^L v_{pl}^W + \sum_{u\in U} D_u^L x_{pu} + LM_p^O}{W_p} = v_p^L \quad \forall p \in P \tag{4.18}$$

$$\frac{\sum_{l\in\mathcal{L}} D_{pl}^V v_{pl}^W + \sum_{u\in U} D_u^V x_{pu} + VM_p^O}{W_p} = v_p^V \quad \forall p \in P \tag{4.19}$$

$$Min^T \leq A_T^W v_p^W + A_T^L v_p^L + A_T \leq Max^T \quad \forall p \in P \tag{4.20}$$

$$Min^{DA} \leq A_{DA}^W v_p^W + A_{DA}^L v_p^L + A_{DA} \leq Max^{DA} \quad \forall p \in P \tag{4.21}$$

$$A_{DF}^W v_p^W + A_{DF}^L v_p^L + A_{DF} \leq Max^{DF} \quad \forall p \in P \tag{4.22}$$

$$A_M^W v_p^W + A_M^L v_p^L + A_M = v_p^M \quad \forall p \in P \tag{4.23}$$

$$v_p^M - v_p^V \geq Min^{GM} \quad \forall p \in P \tag{4.24}$$

The displacement of the vessel is defined by constraint (4.17). Variable v^L is defined by constraint (4.18) using the approximation defined in (4.4). The same approximation is used in constraint (4.19) for the calculation of the Vertical center of gravity (vcg). Note that the approximation on the vcg of location l considers the release and therefore changes at each port p. Constraint (4.20) represents the linearized hydrostatic calculation of trim. Given the displacement of the vessel v^W and its lcg v^L, constraint (4.20) approximates the trim with a plane defined by the coefficients A_T^W, A_T^L, and A_T. The calculated trim is then kept within limits. Changing the coefficients accordingly, constraint (4.21) and (4.22) approximate the aft and fore draft of the vessel. Both drafts are kept within the maximum limits. Due to the propeller it is also necessary to constrain the aft draft to be at a minimum depth. The metacenter is also calculated using the hydrostatic approximation and it is defined in constraint (4.23). The GM is then calculated in constraint (4.24) and kept above the security limit. Below the calculation of the buoyancy force and the constraints that limit shear force and bending moment at different points in the vessel.

$$0.5 D_b^B \sum_{s\in\{b,b+1\}} A_{Bs}^W v_p^W + A_{Bs}^L v_p^L + A_{Bs} = v_{pb}^B \quad \forall p \in P,\, b \in B \tag{4.25}$$

$$W_f^{S,\alpha} + \sum_{l\in\mathcal{L}} G_{lf}^\alpha v_{pl}^W + \sum_{u\in U} G_{uf}^\alpha x_{pu} - \sum_{b\in B} G_{bf}^\alpha v_{pb}^B = v_{pf}^{S\alpha} \quad \forall p \in P,\, f \in F, \alpha \in \{A, F\} \tag{4.26}$$

$$W_f^{B,\alpha} + \sum_{l\in\mathcal{L}} A_{lf}^\alpha G_{lf}^\alpha v_{pl}^W + \sum_{u\in U} A_{uf}^\alpha G_{uf}^\alpha x_{pu} - \sum_{b\in B} A_{bf}^\alpha G_{bf}^\alpha v_{pb}^B = v_{pf}^{B\alpha} \quad \forall p \in P, f \in F, \alpha \in \{A, F\} \tag{4.27}$$

$$Min_f^S \leq G_f v_{pf}^{SF} + (1 - G_f) v_{pf}^{SA} \leq Max_f^S \tag{4.28}$$

$$Min_f^B \leq G_f v_{pf}^{BF} + (1 - G_f) v_{pf}^{BA} \leq Max_f^B \tag{4.29}$$

The buoyancy of the section of a vessel between two adjacent stations is approximated by constraint (4.25). In the shear and bending calculations, we take into account the forces aft or fore of a frame. Since frames do not always coincide with the starting

points of tanks, locations or buoyancy stations, it is necessary for a given frame $f \in F$ (a fixed calculation point), to know the fraction of weight that needs to be taken into account from a location l, tank u, or station b, which is the purpose of the constants $G^A_{\{l,u,b\}f}$ and $G^F_{\{l,u,b\}f}$. Shear force and bending moment are calculated at each vessel frame and must be within limits. Since weight and buoyancy components are computed based on approximations, there is a significant accumulation of error when calculating forces and moments acting at each frame. To tackle this issue, constraints (4.26) and (4.27) compute shear forces and bending moments, respectively, both, wrt. resulting forces acting fore and aft of the frame. Since the fore-based computation is accurate in the bow and the aft-based computation is accurate in the stern, we base a final estimate of the shear forces and bending moments on a proportional blend of the aft and fore computations over the length of the vessel. Thus, the combined shear force is constrained to be within limits by (4.28), while constraint (4.29) does the same for the bending moment. Finally, the constraints that compute the change in ballast water:

$$E_u - x_{pu} \leq y_{pu} \quad \forall p \in P,\, u \in U \tag{4.30}$$

$$x_{pu} - E_u \leq y_{pu} \quad \forall p \in P,\, u \in U \tag{4.31}$$

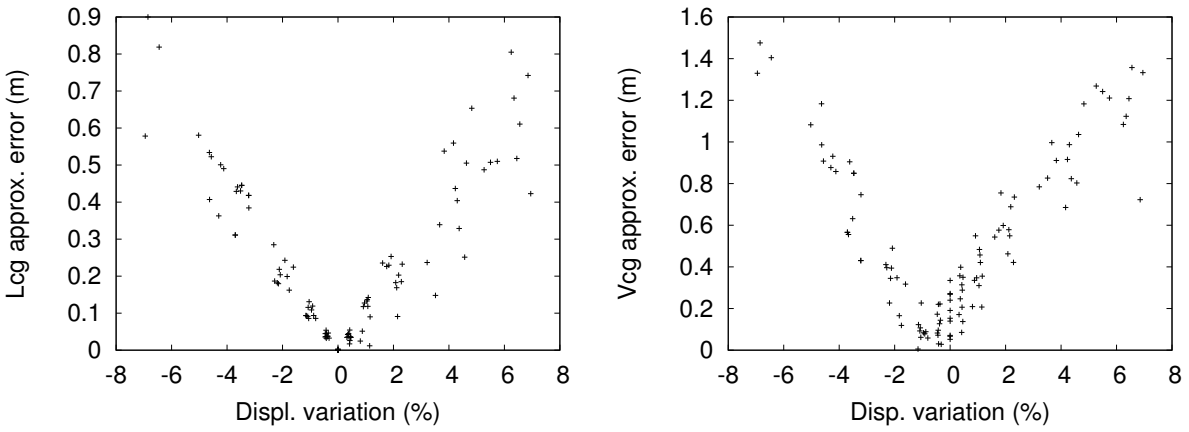
Constraints (4.30) and (4.31) provide a lower bound for the cost variable y_u quantifying the changes in tank configuration from the initial estimate.

4.4 Analysis of Model Accuracy

The model has been evaluated experimentally in a case study involving 10 industrial stowage plans for a vessel of approximately 15.000 TEUs. We constrain ourselves, without loss of generality, to analyse the model for one loading port and several discharge ports, i.e., with p_0 being the current port, we define the set of transport as $TR = \{< p_i, p_j > | p_i = p_0, p_j \in P\}$. Also, without loss of generality, we remove the integrality constraint from the decision variables $x^{t\tau}_l$ and allow the assignment of a continuous number of containers to a location. In Pacino et al. (2011), it has been shown that this decision has a reduced impact on the slot planning phase, while considerably reducing CPU time. The linear approximations used by our now LP model for the generated solutions are compared with exact manual calculations. Table 4.2 gives an overview of the characteristics of the instances. We performed experiments allowing different changes in displacement and observed how the accuracy of the model changed accordingly. First we consider the linear approximation of the center of gravity of the vessel. Figure 4.4 shows two graphs describing how the approximations of the longitudinal (a), and vertical (b) center of gravity behave as the displacement changes. For both graphs, the horizontal axis represents the change of displacement in percent, while the vertical axis represents the error in meters. Each point in the graph is generated by forcing changes in the amount of ballast water of the 10 test instances. In Figure 4.4 (a) it is possible to see, as expected, that when the displacement is unchanged the value of the lcg is accurate, and the more the displacement moves away from its true value, the

Instances Characteristics								
ID	TEU (%)			Weight (%)			Displacement (tons)	Tanks (tons)
	Total	Release	Load	Total	Release	Load		
1	91.79	39.05	52.74	32.05	13.12	18.93	148520	5160
2	73.83	36.66	37.17	45.43	23.14	22.29	176008	8161
3	59.74	18.45	41.29	42.23	12.07	30.16	168696	6952
4	80.98	28.93	52.06	53.01	19.91	33.10	192286	6962
5	65.65	13.38	52.28	25.76	8.00	17.76	135396	10578
6	48.61	18.01	30.60	24.89	8.83	16.06	133489	10578
7	69.27	28.29	40.98	40.86	17.40	23.46	161492	5199
8	45.81	13.16	32.64	28.84	7.77	21.07	143815	9976
9	58.56	24.89	33.67	30.14	13.78	16.36	141487	5239
10	59.04	20.47	38.57	31.97	9.96	22.01	146120	7341

Table 4.2: *Characteristics of the test instances.* Starting from the left the first three columns indicate: the ID of the instance, the total utility percentages in terms of TEU capacity used, the percentage of containers in the release and in the loadlist. The next three columns indicate percentages of utilization in terms of weight, in total, for the containers in the release, and in the loadlist. The initial displacement and the estimated ballast water are given in the last two columns.

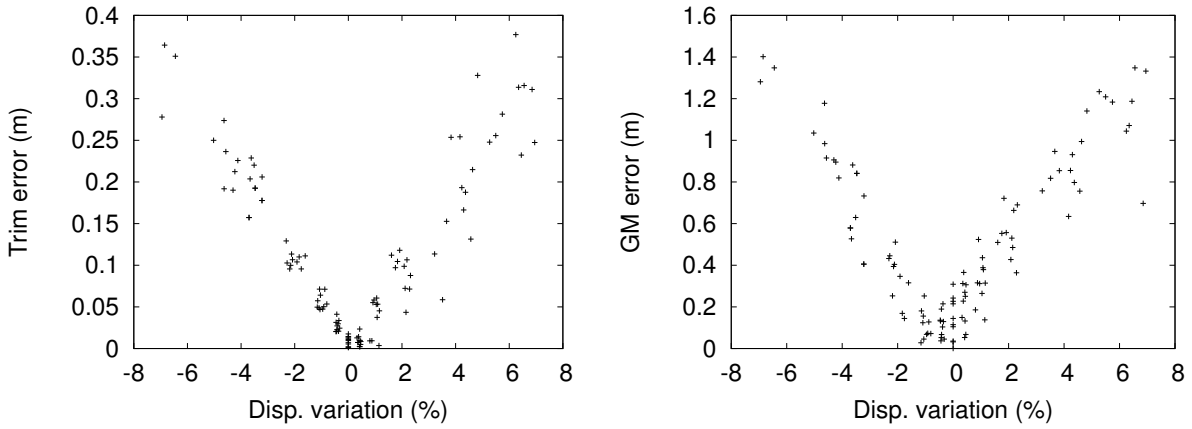


(a) (b)

Figure 4.4: (a) Error in the lcg. (b) Error in the vcg.

less accurate the approximation becomes. Note that for a displacement range of 5%, the calculation inaccuracy is at most 0.3 meters and is thus still accurate enough for practical usage. The calculations for the vcg, however, are not as accurate, (Figure 4.4 (b)). Within the 5% range, the linearized value is at most 0.8 meters from the correct one. This was an expected result, as it is not possible to precisely estimate the vcg of locations since we do not know where the containers will be stowed. The accuracy error of the vcg is, however, still very small.

Now that we have shown that linearizations for the center of gravity are accurate, we focus on analyzing the accuracy of the hydrostatic data linearization. In Figure 4.5



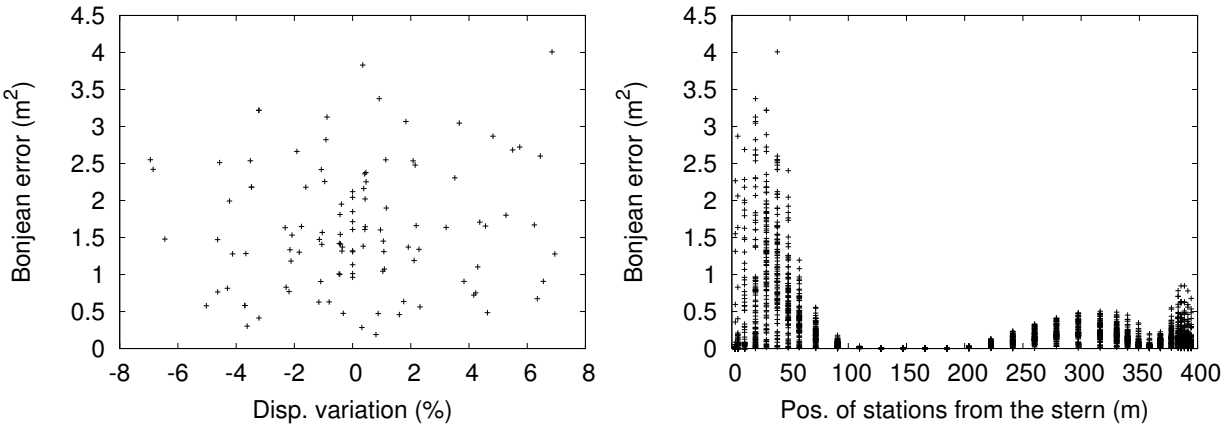
(a) (b)

Figure 4.5: (a) Error for the trim (b) Error for the GM.

we use the same graph as before, with the horizontal axis describing the percentage displacement changes and the vertical axis the calculation error. Figure 4.5 (a) depicts the error for the trim which, as it can be seen, is very small. Within a 5% displacement range the error is, at most, 15 centimeters. Figure 4.5 (b) shows the same analysis for the calculation of GM. Notice that for both calculations, the trim and GM, an error is present even at constant displacement. The error we see is due to the linearization of the hydrostatic functions. For GM it also includes the approximation error of the vcg.

We now move our focus to the linearization of the bonjean. Due to the changes in the shape of the hull in the extremities of the vessel we expect the bonjean calculations to be the most inaccurate in the sections close to the bow and stern, while behaving more smoothly towards midship. Figure 4.6 (a) shows the same analysis we have done before applied to the bonjean areas. The graph shows the maximum error over all bonjean areas as a function of the displacement changes. As can be seen, the variation in displacement is not the main source of error. Most of the inaccuracy is due to the linearization of the hydrostatic data. Figure 4.6 (b) shows how the bonjean error is concentrated at the extremities of the vessel where the hull changes most. The horizontal axis represents the position of the station on the vessel (where 0 is at the bow), and the vertical axis is the bonjean error. Notice that the largest errors are found for stations at the bow. This can be explained by the fact that the range of drafts in our test data forces the linearization of these bonjean areas to be right by the inflection point of the hydrostatic curve (see figure 4.3 (a)), where the linearization is most inaccurate. Better approximations can then be expected for larger drafts, as it is the case for the bonjean of the stations at the stern. The inaccuracy of the bonjean is, however, still quite small if we consider that in the worst case scenario, there is an error of only 4 square meters over an area of over 111 square meters.

Shear forces and bending moment calculations depend on the buoyancy of the ves-

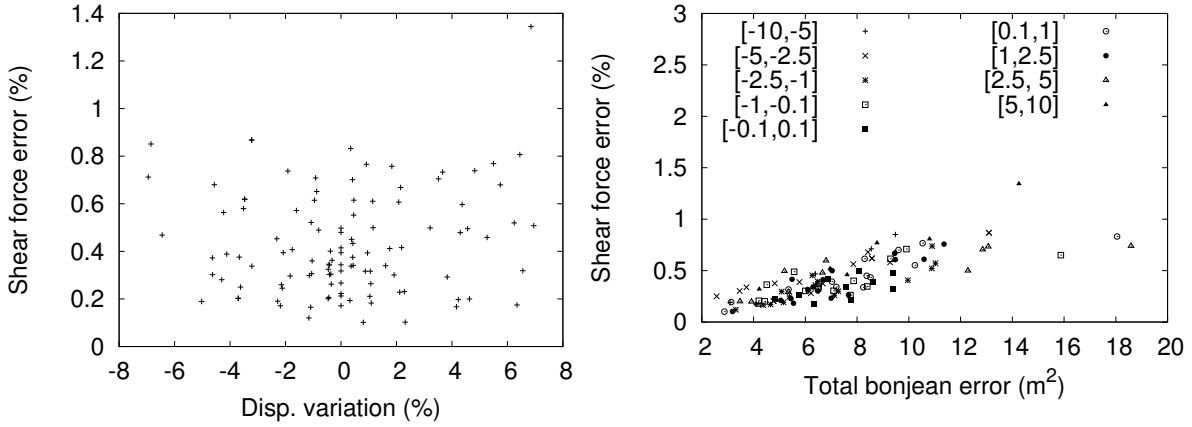


(a) (b)

Figure 4.6: (a) Approximation error of the total bonjean as a function of displacement change. (b) Approximation error of the bonjean areas per station.

sel, which in turn is calculated using the bonjean approximations. Figure 4.7 (a) shows, in the same way as the other graphs, how the percentage error in the shear calculation (the vertical axis), behaves as a function of the variation of the displacement, (the horizontal axis). As expected, the dominant error is not the approximation of the center of gravity of the vessel, since the inaccuracy is more or less the same independent of how much the displacement changes. A tighter relation can be seen when the shear calculation is related to the error in the bonjean linearization. Figure 4.7 (b) shows the percentage error of the shear force calculation as a function of the total error in the bonjean area from the hydrostatic linearization, with the data points grouped according to their percentage of displacement variation. As depicted, the error in the shear calculation increases with the error in the bonjean linearization for all the displacement variations. Moreover, for the data points with little or no displacement change (displacement variation between $[-0.1, 0.1]$), we can see that the tendency remains the same.

Figure 4.8 (a) shows the error in the bending calculation as a function of the changes in displacement. As expected, similarly to the shear force calculation, there is no clear tendency indicating a relation between the error in the bending calculations and the approximation of the lcg. With respect to the impact of bonjean error in bending calculations, we expect a similar tendency to that of Figure 4.7 (b), since bending moment and shear force calculations depend similarly on the vessel buoyancy. Moreover, due to the fact that the bonjean error is amplified by the multiplication of the arm, it is interesting to look at the error at each calculation frame in order to see how the error of the bending moment changes. We do this by forcing the displacement to remain constant, thus removing the error of the lcg approximation, and analyzing how the bending error changes at each calculation frame as the draft changes. The result is shown in Figure 4.8 (b), where the horizontal axis represents the frames of the vessel



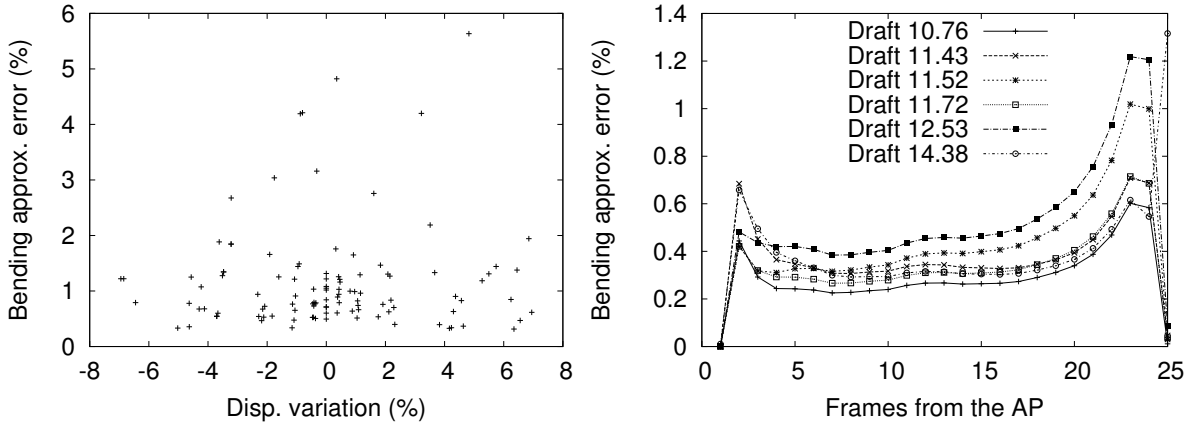
(a) (b)

Figure 4.7: Approximation error of the shear forces as (a) a function of displacement change, and (b) a function of the total bonjean error with data points grouped by the % of displacement variation.

and the vertical axis is the percentage of error in the bending moment calculation. Each of the lines plotted in the graph represents one of the 6 test instances selected for this experiment, each of which has a different draft. The 4 instances not included in the experiment had a similar draft and followed the same tendency as that illustrated by the instances plotted in Figure 4.8 (b). According to the bending calculations in our model, the bending error should be low in frames near the vessel ends since the most accurate bending calculations are chosen at those points, while the error should increase close to midship, since at this point the arm length reaches its highest value and amplifies the bonjean error. This, however, is not the tendency depicted in Figure 4.8 (b), where the bonjean error seems to even out as it moves away from the vessel ends, even after being amplified by the arm. This indicates that the impact of the arm in the bending error is considerably reduced by the bending calculation introduced in this chapter.
One more thing worthy of notice is that there is no direct relation between the bending error and the draft of the vessel. This is due to the non-linear shape of the hull. The bending error for constant displacement does not exceed 1.4% (see Figure 4.8 (b)), however, when variable displacement is considered, errors of up to 5% might be reached within a 5% displacement variation, (see Figure 4.8 (a)). We consider these approximations acceptable. Higher accuracy can be achieved by reducing the linearization error of the hydrostatic functions for the bonjean.

4.5 Conclusion

This chapter introduced a mathematical programming model including ballast tanks for the master planning phase of a 2-phase stowage planning optimization approach.



(a) (b)

Figure 4.8: Approximation error of the bending moment, (a) as a function of displacement change, and (b) at each bonjean station for constant displacement.

Analysis of 10 real instances show that our model is successful at coping with variable displacement with acceptable error tolerance. Within a 5% band of the current displacement, the master plans generated were seaworthy, with the error in stability calculations increasing proportionally to the variability of the displacement. In future work, we plan to introduce piece-wise linearizations of the bonjean hydrostatic data to reduce the error in the stress calculations in our model. This must be done carefully since it might be necessary to include boolean variables that will negatively impact the performance of the solver.

Chapter 5

Heuristic Optimization for Interactive Stowage Planning

Chapter 3 and 4described models for the automatic generation of stowage plans. They correspond to the second and first phase, respectively, of a 2-phase hierarchical decomposition described in chapter 1, (see Figure 1.3). In this chapter, we present a heuristic approach used in *Angelstow*[1], a commercial decision support tool for the interactive generation of stowage plans based on stowage coordinators' know-how. This approach mimics the 2-phase decomposition mentioned above, incorporating stowage coordinators' expert knowledge to tackle important objectives (e.g., crane split, overstowage), and constraints (e.g., stability and stresses forces), into an LP model to solve the master planning part (first phase), while using a greedy algorithm to solve the slot planning part (second phase). To guarantee a smooth flow of information with the stowage coordinator, stowage plans are generated within a few seconds. This chapter is based on Delgado et al. (2012).

5.1 Introduction

Liner shipping companies are in their infancy using optimization-based decision support tools for stowage planning. Most of the research on stowage planning during the past twenty years has focused on algorithms to automatically generate stowage plans (e.g., Gumus et al. (2008); Ambrosino et al. (2010); Yoke et al. (2009); Pacino et al. (2011)), limiting the role of expert users to result validation. Though some of these approaches generate good quality stowage plans, difficulties arise using them in practice. Stowage planning is full of details disregarded in the models solved by automatic approaches. Stowage coordinators need to be able to easily adapt automated plans to specific situations, or be directly involved in their generation for these plans to be successfully used in practice.

Angelstow is a decision support tool that assists stowage coordinators during the

[1]Ange Optimization ApS, available for purchase at www.http://ange.dk/main/angelstow.

planning process. It provides an interface for defining *planning scenarios* (i.e., a set of high and low level preferences that define the backbone of a stowage plan), and the capability of turning these scenarios into stowage plans. Using these features, stowage coordinators can stow vessels interactively. A coordinator devises a planning scenario that Angelstow turns into a stowage plan, providing feedback into the impact of the coordinators' preferences. If the resulting stowage plan is not satisfactory, stowage coordinators can change their scenario and let Angelstow generate a new stowage plan. To avoid that this interactive process jeopardizes the usability of the tool, fast stowage planning is a key factor. A desired runtime for a stowage planning algorithm in this setup is at most two seconds on average.

In this chapter, we introduce a 2-phase heuristic optimization approach that generates stowage plans for containers to be loaded at a single port. This approach mimics the decomposition depicted in figure 1.3. In the master planning phase[2] the planning scenario is incorporated into an LP model that assigns containers to locations, while in the slot planning phase a greedy algorithm stows the containers assigned to each location into vessel slots. We evaluate our heuristic optimization approach on three real-life stowage plans. Each stowage plan corresponds to the stowage condition of a large vessel of approximately 15,000 TEUs after leaving the last of six consecutive ports, with the vessel being empty in the first port. To carry out our experiments, we extract the planning scenario that a stowage coordinator devised at each port and then let our approach generate stowage plans for all ports. In this way, we achieve a fair comparison between the stowage plans generated by our approach and those generated by the stowage coordinators.

Our results show that the runtime of our approach is reasonable, (1.27 seconds on average). Moreover, it rolls out on average 0.58% and at most 1.89% of the containers to be transported, and utilizes stack heights as efficient as the stowage coordinators. Additionally, we evaluate the impact of producing master plans with an LP model, which implies that a continuous number of containers might be assigned to locations, something physically impossible. The LP model is transformed into an IP model by adding the corresponding integrality constraints. We solve the IP model with an optimality gap of 1% and 5% and generate stowage plans based on these new master plans. Then we compare the results with the original plans generated based on the LP master plans. Our experiments show that for our set of instances there is no dominance from stowage plans based on the IP model with respect to those based on the LP model, while the solution time considerably increases. The remainder of the chapter is organized as follows: Section 5.2 describes the problem. Section 5.3 introduces Angelstow. Section 5.4 describes related work and section 5.5 presents our heuristic optimization approach. Finally, Section 5.6 and 5.7 present the experiments and draw conclusions.

[2]Note that the master planning phase of the heuristic introduced in this chapter differs from the one presented in chapter 4. It neither includes specific constraints for stability and stress forces nor consider ballast tanks.

5.2 Problem Definition

In this chapter, we focus on the generation of stowage plans based on planning scenarios defined by stowage coordinators. A planning scenario encapsulates a set of high level preferences, i.e., selection of a single discharge port for all containers stowed in a location and upper bounds for the weight utilization of bays, and low level preferences, i.e., requiring that certain containers are stowed in certain vessel slots. They embed the know-how of stowage coordinators with respect to how to tackle complicated combinatorial objectives and constraints, and how to stow containers with special demands, e.g., IMO and Pallet-wide. The high level preferences focus on minimizing hatch overstowage to avoid extra crane moves, and on the even distribution of moves over the length of the vessel to minimize the makespan of the quay cranes. Moreover, they must ensure seaworthiness of the vessel with respect to shear force and bending moments limits, as well as line-of-sight, trim, draft and GM requirements.

Master planning assigns containers to locations following the planning scenario devised by the stowage coordinator and satisfying the different capacities of each location according to the features containers may have, e.g., reefer containers limited by the available reefer plugs, volume capacity of *20'* and *40'* containers. A feasible slot plan must follow the assignment of containers specified in its corresponding master plan and form container stacks that satisfy stacking rules and constraints. A stowage plan must satisfy the rules specified in section 3.2 for the slot planning model presented in chapter 3, together with two extra rules: the planning scenario must be respected and no overstowage is allowed.
In addition, we follow the set of rules of thumb listed below to generate high quality stowage plans:

- Containers are distributed evenly among stacks on deck
- The number of used stacks is minimized in storage areas below deck
- Wasted volume capacity is minimized
- It is preferred to stow containers below deck, if possible
- Heavy containers should go below deck
- Preferential loading of reefer containers over dry containers
- Heavy containers cannot be stowed on top of light containers on deck

5.3 Angelstow

Angelstow provides an interface that allows stowage coordinators to define planning scenarios. In these scenarios, a stowage coordinator defines the discharge port of the containers to stow in a location. Only a single discharge port is allowed for each

location. This is seldom a limitation in practice since most vessels are split into sufficient locations to allow only stowing containers of one discharge port. Moreover, due to the robustness of this approach with respect to avoiding overstowage in downstream ports, it is considered good practice by stowage coordinators.[3]

Automatic stowage planning can be activated at any time, but only the locations for which the stowage coordinator has defined a discharge port will be stowed. The stowage coordinator can also set upper bounds on the total weight of containers stowed in a bay. In this way, scenarios only covering vessel sections for which a discharge port has been specified can be stowed without the stowage coordinator losing control of the overall weight distribution over the vessel. This is essential for achieving the desired draft, trim, GM, and stress forces in the final stowage plan.

Recall that a stowage plan is an assignment of the containers to load in a port to specific vessel slots. To validate a planning scenario, that is, to verify whether the stowage coordinator's preferences can be turned into a valid stowage plan, the stowage coordinator must generate an actual stowage plan that follows the preferences specified in the scenario. This implies stowing a considerable number of containers for testing each scenario. Stowage coordinators generate plans under time pressure and uncertainty, since they receive the loadlist of containers a few hours before the vessel calls port. Due to these conditions, the number of planning scenarios that can be analyzed is very limited. To address this issue, Angelstow provides stowage coordinators with the capability of automatically turning these scenarios into stowage plans. The generation of a stowage plan then becomes the interactive process described next. First, the stowage coordinator deviss a planning scenario. Next, Angelstow generates a stowage plan based on the scenario specified by the stowage coordinator. The stowage coordinator can modify the plan that Angelstow generated by manually re-stowing containers. In case it is not possible to adapt the stowage plan to what the stowage coordinator considers a reasonable solution, the planning scenario can be changed and the process starts over again. To avoid this interactive process, to jeopardize the usability of the tool, fast stowage planning is a key factor, which is a concern addressed by the heuristic optimization approach presented in Section 5.5. We estimate that a time limit of two seconds for generating stowage plans is necessary to efficiently support interactive optimization in Angelstow.

A multi-port view of stowage planning is important to reduce the negative impact of a stowage plan in downstream ports. A bad stowage plan can considerably reduce the vessel capacity for containers to load in upcoming ports. Angelstow provides a multi-port view (see Figure 5.1), where stowage coordinators can see the effect of their decisions in downstream ports. Additionally, even though stowage coordinators focus their work on the current port, information available in loadlists for downstream ports (i.e., forecasts), can be used to define planning scenarios for these ports as well.

When a stowage coordinator wants to check the feasibility of the scenarios formu-

[3]A disadvantage of following this approach rigidly is a potentially less efficient usage of the volume and weight capacity of the vessel.

Figure 5.1: Front view of Angelstow. To the left, a summary of the loadlist of containers and a view of the current state of the ballast tanks. In the center, the multi-port view of Angelstow. Each row represents a port where the vessel calls. To the right, the stresses graphs and weight distribution along the vessel bays.

lated for each of the ports a vessel calls, Angelstow generates stowage plans for each port independently using the approach introduced in section 5.5. Figure 5.2 depicts the process that Angelstow follows to generate stowage plans for all the ports the vessel calls. At each port, the stowage planning component uses as input 1) the planning

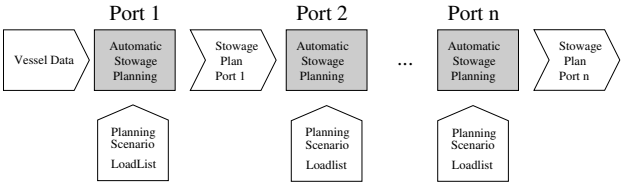


Figure 5.2: Multi-port stowage planning.

scenario defined by the stowage coordinator from Angelstow, and 2) a loadlist of containers for the port. The containers stowed by Angelstow in the previous ports have fixed positions and become part of the input for the next port. Thus, the stowage plan made for the current port translates into a list of release containers for the next downstream port. A more sophisticated approach to multi-port stowage planning would be to represent the problem in a single optimization model, as done in chapters 3 and 4. This has not been done since 1) the impact is limited because most of the important

decisions with respect to multi-port planning are part of the planning scenario, and 2) this may deteriorate the runtime performance.

5.4 Literature Review

Academic work on stowage planning can be divided into two main categories: single-phase and multi-phase approaches. Single-phase approaches have a plain representation of the stowage planning problem, (e.g., stow containers into vessel slots). A common denominator among these approaches is to sacrifice model accuracy to achieve scalability. Early work in this category (e.g., Scott and Chen (1978); Aslidis (1984)), introduced heuristic approaches that solved simplifications of the stowage planning problem. These approaches sequentially refine solutions by applying placement heuristics, local search, and solving IP models. Later on, Botter and Brinati (1992), introduced an IP model of the first accurate but intractable formulation of the stowage planning problem. Constraint programming (Ambrosino and Sciomachen (1998)), metaheuristics such as genetic algorithms (e.g., Davidor and Avihail (1996); Dubrovsky et al. (2002)), and simulated annealing (Flor (1998)), and more recently IP (e.g, Ambrosino and Sciomachen (2003); Giemsch and Jellinghaus (2004); Li et al. (2008)), have been used to solve still simplified versions of the stowage planning problem. A case-based reasoning approach is proposed in Nugroho (2004), where stowage plans are generated by remembering how similar past plans were made. The system suggests possible solutions to the stowage coordinators that can be turned into stowage plans.

Multi-phase approaches decompose the problem hierarchically into two or more layers of abstraction, (e.g., first assign containers to bay sections, then stow them into vessel slots). These approaches are currently the most successful in terms of model accuracy and scalability. The first work of this type appeared in the early 1970s, where Webster and Van Dyke (1970), introduced a 3-phase heuristic. Twenty years later, a model that includes several major aspects of the problem in a 2-phase approach was introduced in Wilson and Roach (1999). This approach solves the stowage planning problem for multiple ports. It uses a branch-and-bound algorithm for solving the first phase, where containers are assigned to sub-sections of the vessel, and applies a tabu search for stowing containers into vessel slots in the second phase. Other approaches that use similar decompositions include solving multi-port stowage planning with an iterative improvement approach. This is based on the transportation simplex method (Kang and Kim (2002)), a bin-packing heuristic (Zhang et al. (2005)), and solving a mixed-integer program for the first phase, and constraint programming and constraint-based local search models for the second one, (Pacino et al. (2011)). 3-phase approaches include combinations of constructive heuristics, 0/1 IP, metaheuristics (e.g., Ambrosino et al. (2010)), and heuristics combined with LS, (e.g., Yoke et al. (2009)).

Multi-phase approaches developed by the industry include a multi-stage placement heuristic using a number of lower-bounds (Gumus et al. (2008)), and a combination of sequential LPs for assigning containers to bay sections, and a hierarchy of IPs for

stowing them into vessel slots (Guilbert and Paquin (2010)).

Several software tools are available on the market for container stowage planning decision support, (e.g., Navis (2013); INTERSCHALT maritime systems AG (2013); Muller+Blanck Software GmbH (2013); Autoship Systems Corporation (2013)). All of them assist stowage coordinators with the generation of stowage plans by providing feedback on partial stowage planning decisions. To the best of our knowledge, our heuristic is the first attempt to use optimization techniques for interactive generation of stowage plans.

5.5 Heuristic Optimization Approach to Stowage Planning

In this section, we introduce a stowage planning heuristic based on the 2-phase decomposition presented in chapter 1. This heuristic stows containers following a planning scenario devised by a stowage coordinator using Angelstow, and fulfills and optimizes all constraints and objectives specified in section 5.2. In the master planning phase, containers are grouped together by their features into types and assigned to locations by an LP model. In the slot planning phase, a greedy algorithm finds specific slots for all containers following the assignment made by the master planning phase. Currently, we limit our scope to consider containers that are *20'* and *40'* long and model reefer and high-cube characteristics.

5.5.1 LP Model for Master Planning

Here we present the LP model that solves the master planning phase of our heuristic optimization approach. Containers are grouped into types based on their features. Recall that a type $\tau \in \mathcal{T}$ is a 4-tuple, (l, h, r, w). For the model presented in this section we define $\mathcal{T} = \{(l, h, r, w) \,|\, l \in L = \{20, 40\}, h \in H = \{DC, HC\}, r \in R = \{RF, NR\}, w \in W\}$, where HC refers to high-cube containers and $w \in W$ is the weight class of the container defined by the weight of the container rounded to the nearest integer ton. The LP model assigns types of containers loaded on a single port to locations. Though all containers are loaded in a single port they have multiple discharge ports. With $p' \in P$ being the loading port (i.e., the port where the containers are currently being stowed), we define the set of transports in our model as $TR = \{< p_i, p_j > | p_i = p', p_j \in P\}$.

The decision variables $x_l^{t\tau} \in \mathbb{R}^+$ represent the number of containers of type $\tau \in \mathcal{T}$ stowed in location $l \in \mathcal{L}$ during transport t. In this formulation we have removed the integrality constraint on the decision variables. Given the number of containers that can be stowed in a location (up to a couple of hundred), and the small number of valid types per location seen in practice, the loss of precision in our solution by doing this is low and can be easily dealt with in the second phase of our heuristic. We test this assumption in the experimental section.

We remind the reader that the sets and constants used in the model presented in this chapter have been previously defined in the section *Sets and Constants*, at the beginning of this document. Below we introduce the variables of our model.

Decision Variables

$x_l^{t\tau} \in \mathbb{R}^+$	Containers of type τ in location l during transport t.

Auxiliary Variables

$y_{p'l}^{HC} \in \mathbb{R}^+$	Number of high-cube containers stowed in l above the high-cube killing limit at port p'.
$y_{p'}^{O} \in \mathbb{R}^+$	Number of misused odd slots at port p'.
$v_{p'l}^{\neg RO} \in \mathbb{R}^+$	*40'* volume capacity gained by stowing *20'* non-reefer containers in the non-reefer odd slots of location l at port p'.
$v_{p'l}^{OR} \in \mathbb{R}^+$	*40'* volume capacity gained by stowing *20'* containers in the reefer odd slots of location l at port p'.
$y_{p'l}^{O} \in \mathbb{R}^+$	Number of odd slots not used in l at port p'.
$y_{p'}^{O} \in \mathbb{R}^+$	Number of odd slots not used in the vessel due to misplacement of *20'* containers in port p'.

We propose the LP model below for solving the problem of assigning container types to locations in the first phase of our heuristic. First we introduce the objective function of our model.

$$maximize$$

$$\sum_{t \in TR_{p'}^{ON}} \sum_{\tau \in \mathcal{T}} \sum_{l \in \mathcal{L}} C_l^{\tau} x_l^{t\tau} - C^{HC} \sum_{l \in \mathcal{L}} y_{p'l}^{HC} - C^{OD} y_{p'}^{O} \tag{5.1}$$

The objective (5.1) maximizes the quality of the stowage plan. Each decision variable $x_l^{t\tau}$ has a cost C_l^{τ} associate to them that reflects preferences defined by our industrial collaborator with respect to container types and locations. For instance, reefer containers are more profitable, therefore, we prefer to load reefer over non-reefer containers. Locations below deck are preferred over locations on deck due to accessibility, and heavy containers are preferred to be stowed below deck to increase stability of the vessel.
Recall that high-cube containers utilize more volume capacity than dry cargo containers. When the number of high-cube containers stowed in location l goes beyond $K_{p'l}^{K}$, the slot capacity of l is reduced in order to provide the extra volume capacity necessary to stow the additional high-cube containers. The cells involved in this slot capacity reduction are known as *killed cells*, i.e., cells that have been partially utilized by high-cube containers and where it will not be possible to stow any additional container. We penalize the number of high-cube containers stowed in location l above $K_{p'l}^{K}$ with cost C^{HC}. The number of misused odd slots in the vessel is penalized by cost C^{OD}. Odd slots are at the bottom cells of a stack and it is necessary to fill them up with *20'* containers in order to access the full capacity of the stack. Our model penalizes odd slots that are not being used when there are *20'* containers available in the loadlist to do so. Next, we introduce the constraints representing the planning scenario defined by the stowage coordinator.

$$s.t.$$

$$\sum_{t \in TR^{l}_{p'}} \sum_{\tau \in \mathcal{T}} x_l^{t\tau} = 0 \quad \forall l \in \mathcal{L} \tag{5.2}$$

$$\sum_{l \in \mathcal{L}} x_l^{t\tau} \leq LD_t^{\tau} \quad \forall t \in TR,\, \tau \in \mathcal{T} \tag{5.3}$$

$$\sum_{t \in TR^{ON}_{p'}} \sum_{l \in \mathcal{L}_v} \sum_{\tau \in \mathcal{T}} W_\tau x_l^{t\tau} \leq W^{U}_{p'v} \quad \forall v \in V \tag{5.4}$$

Note that all capacities in the model are reduced by the release. Constraint (5.2) restricts the containers to stow in location l by removing containers with a different discharge port to that chosen by the stowage coordinator in the planning scenario. Constraint (5.3) ensures that we do not stow more containers of type τ on transport t than those available from the loadlist. Constraint (5.4) restricts the weight of a bay according to the stowage coordinator's preferences specified in the planning scenario. Below we present the constraints necessary to model odd slots and their impact on the volume capacity of the locations.

$$\sum_{t \in TR^{ON}_{p'}} \sum_{\tau \in \mathcal{T}^{20} \cap \mathcal{T}^{NR}} A^{\neg R}_{p'l} x_l^{t\tau} \geq v^{\neg RO}_{p'l} \quad \forall l \in \mathcal{L}^O \tag{5.5}$$

$$v_l^{\neg RO} \leq A^{\neg R}_{p'l} K^{\neg RO}_{p'l} \quad \forall l \in \mathcal{L}^O \tag{5.6}$$

$$\sum_{t \in TR^{ON}_{p'}} \sum_{\tau \in \mathcal{T}^{20}} A^{R}_{p'l} x_l^{t\tau} \geq v^{OR}_{p'l} \quad \forall l \in \mathcal{L}^O \tag{5.7}$$

$$v_l^{OR} \leq A^{R}_{p'l} K^{OR}_{p'l} \quad \forall l \in \mathcal{L}^O \tag{5.8}$$

$$\sum_{t \in TR^{ON}_{p'}} \sum_{\tau \in \mathcal{T}^{20}} x_l^{t\tau} \geq \frac{v^{\neg OR}_{p'l}}{A^{\neg R}_{p'l}} + \frac{v^{OR}_{p'l}}{A^{R}_{p'l}} \quad \forall l \in \mathcal{L}^O \tag{5.9}$$

When present, odd slots are mainly found at the bottom cells of a stack. For the *40'* volume capacity of a stack with odd slots to become available, the odd slots must be filled in with *20'* containers. Thus, in locations with odd slots *40'* volume capacity is increased proportionally to the utilized odd slots. Notice that we must distinguish between *20'* reefer and non-reefer containers since a reefer container can only be placed in reefer odd slots. Constraint (5.5) sets the lower bound of the auxiliary variable $v^{\neg RO}_{pl}$ to be the extra *40'* volume capacity gained by stowing *20'* non-reefer containers in non-reefer odd slots of location l. An upper bound to this gain is provided by constraint (5.6). In a similar fashion, constraints (5.7) and (5.8) provide lower and upper bounds, respectively, for the auxiliary variable v^{OR}_{pl}. The lower bound is calculated as the extra *40'* volume capacity gained by stowing *20'* containers (reefer and non-reefer), in the reefer slots of l, while the upper bound is the maximum *40'* volume gain when stowing *20'* containers in all reefer odd slots. To avoid counting volume gains produced by the *20'* non-reefer containers twice, constraint (5.9), forces the number of *20'* containers to be at least as many as the containers used to calculate the volume gains when stowing

20' in odd slots. Next, we introduce the constraints that limit volume, weight, and reefer capacity of the locations.

$$\sum_{t \in TR_{p'}^{ON}} \sum_{\tau \in \mathcal{T}} \Pi_{\tau}^{TEU} x_l^{t\tau} \leq K_{p'l}^{TEU} \quad \forall l \in \mathcal{L} \tag{5.10}$$

$$\sum_{t \in TR_{p'}^{ON}} \sum_{\tau \in \mathcal{T}^{20}} x_l^{t\tau} \leq K_{p'l}^{20} \quad \forall l \in \mathcal{L} \tag{5.11}$$

$$\sum_{t \in TR_{p'}^{ON}} \sum_{\tau \in \mathcal{T}^{R}} x_l^{t\tau} \leq K_{p'l}^{RF} \quad \forall l \in \mathcal{L} \tag{5.12}$$

$$\sum_{t \in TR_{p'}^{ON}} \sum_{\tau \in \mathcal{T}^{R}} \Pi_{\tau}^{FEU} x_l^{t\tau} \leq K_{p'l}^{RC} \quad \forall l \in \mathcal{L} \tag{5.13}$$

$$\sum_{t \in TR_{p'}^{ON}} \sum_{\tau \in \mathcal{T}^{20}} W_{\tau} x_l^{t\tau} \leq W_{p'l}^{20} \quad \forall l \in \mathcal{L} \tag{5.14}$$

$$\sum_{t \in TR_{p'}^{ON}} \sum_{\tau \in \mathcal{T}} \Pi_{\tau}^{FEU} W_{\tau} x_l^{t\tau} \leq W_{p'l}^{40} \quad \forall l \in \mathcal{L} \tag{5.15}$$

$$\sum_{t \in TR_{p'}^{ON}} \sum_{\tau \in \mathcal{T}} \Phi_{\tau} x_l^{t\tau} \leq K_{p'l}^{V} \quad \forall l \in \mathcal{L} \tag{5.16}$$

$$\sum_{t \in TR_{p'}^{ON}} \sum_{\tau \in \mathcal{T}^{40}} \Phi_{\tau} x_l^{t\tau} \leq K_l^{40N} + v_l^{\neg RO} + v_l^{OR} \quad \forall l \in \mathcal{L}^{O} \tag{5.17}$$

$$\sum_{t \in TR_{p'}^{ON}} \sum_{\tau \in \mathcal{T}^{40}} \Phi_{\tau} x_l^{t\tau} \leq K_{p'l}^{40V} \quad \forall l \in \mathcal{L} \backslash \mathcal{L}^{O} \tag{5.18}$$

$$\sum_{t \in TR_{p'}^{ON}} \sum_{\tau' \in \mathcal{T}^{(W_{\tau'} \leq W_{\tau})} \cap \mathcal{T}^{(\Pi_{\tau'}^{TEU} = \Pi_{\tau}^{TEU})}} x_l^{t\tau'} \leq K_{p'l}^{\tau} \quad \forall l \in \mathcal{L}, \tau \in T \tag{5.19}$$

Constraints (5.10)-(5.13) and (5.14)-(5.15) are equivalent to constraints (4.6)-(4.9) and (4.11)-(4.12) from section 4.3.1, respectively, and will not be discussed any further. Containers consume volume capacity from locations depending on their type. For instance, *20*' containers need a single slot, while *40*' containers need two. All high-cube containers are *40*' and they consume slightly more than two slots. Constraint (5.16), restricts the volume of containers stowed in $l \in \mathcal{L}$ to be within its volume capacity limit at port p.

We constrain the *40*' volume capacity of locations without odd slots by means of constraint (5.18). Note that this constraint considers volume consumption instead of TEU consumption since *40*' can be dry or high-cube containers. Constraint (5.17), limits the number of *40*' containers in odd locations to be within the *40*' volume capacity. The *40*' volume capacity of location $l \in \mathcal{L}^{O}$ is defined as the volume capacity of the location stacks with no odd slots plus the increment in *40*' volume capacity given by stowing *20*' containers in the odd slots of l. We compute the upper bound

K^{τ}_{pl} in a pre-processing stage. This upper bound is defined as the maximum number of containers of type τ that can be stowed in l when relaxing all constraints but those enforcing weight and TEU capacity. Thus, the number of containers of types with the same length as τ (i.e., demanding the same TEU capacity), and with greater or equal weight must be smaller or equal to K^{τ}_{pl}. Constraint (5.19) enforces this. Finally, the constraints that compute lower bounds for the misused odd slots in the vessel and the high-cube containers above the high-cube killing limits.

$$\sum_{\tau \in \mathcal{T}^{20}} x^{t\tau}_{l} + y^{O}_{p'l} \geq K^{O}_{p'l} \quad \forall l \in \mathcal{L}^{O} \tag{5.20}$$

$$\sum_{t \in TR} (\sum_{l \in \mathcal{L}^{t}} y^{O}_{p'l} - K^{O20}_{t}) \leq y^{O}_{p'} \tag{5.21}$$

$$\sum_{\tau \in \mathcal{T}^{HC}} x^{\tau t}_{l} - y^{HC}_{p'l} \leq K^{K}_{p'l} \quad \forall l \in \mathcal{L} \tag{5.22}$$

Constraint (5.20) provides a lower bound for the number of odd slots not used in locations with odd slots available. In this constraint we assume that *20'* containers will always be stowed in odd slots if they are available. A lower bound for the number of odd slots not used in the vessel due to misplacement of *20'* containers is provided by constraint (5.21). This lower bound is the difference between the odd slots not currently being used, and the odd slots that are known in advance but cannot be used due to the lack of *20'* containers. As mentioned before, when the number of high-cube containers in a location l surpasses the high-cube killing limit it is necessary to reduce the slot capacity of the location in order to stow the extra high-cube containers. The relation between the number of killed cells and the extra number of high-cube containers that can be allocated in l is not linear, i.e., several high-cube containers can be stowed in l by killing a single cell. Any linearization attempt on measuring the slot capacity reduction due to killed cells in l, can be reduced to measure the number of high-cube containers above $K^{K}_{p'l}$ stowed in l. Constraint (5.22) provide a lower bound for auxiliary variable $y^{HC}_{p'l}$.

5.5.2 Greedy Algorithm for Slot Planning

We introduce the greedy Algorithm 1 to stow containers in specific slots (i.e., slot planning) in such a way that the set of stacking constraints and rules of thumb described in Section 5.2 are satisfied and optimized, respectively. The algorithm works as follows. Given a set of containers C and a set of stacks S as parameters, for each container $c \in C$ the algorithm finds the best stack $s \in S$ where c can be stowed, (lines 5-9). Containers are stowed bottom-up in stacks, therefore, when looking for a feasible stack for c, the greedy algorithm only checks the lowest empty cell of each stack. A stack s_i is preferred over stack s_j for stowing container c if by stowing container c at the lowest empty cell of stack s_i more of the rules of thumb described in Section 5.2 are satisfied than by stowing c at the lowest empty cell of stack s_j. Once the best stack has been found, c is

Algorithm 1 GreedySelection(C, S)

```
1:  uc ← ∅ // Set of unstowed containers from C
2:  repeat
3:      Select container c from C
4:      bs ← NULL // Best stack for container c
5:      for s_i ∈ S do
6:          if it is possible to stow c in s_i then
7:              update bs to s_i if s_i is better than bs
8:          end if
9:      end for
10:     if there is no feasible stack for c then
11:         uc ← uc ∪ {c}
12:     else
13:         Stow container c in stack bs
14:     end if
15:     C ← C − {c}
16: until C ≠ ∅
17: return uc
```

stowed in the lowest empty cell of the stack, (line 13). In case it is not possible to find a feasible stack to stow c, the container is added to the list of unstowed containers, uc, (line 11). Once all containers have been stowed, or attempted to be stowed, the list of unstowed containers is returned.

It is important to notice that if the stowing of container c in the lowest empty cell of stack s fulfills all stacking rules and constraints from Section 5.2 , such stowage will not make infeasible any of the previous stows of containers in cells of stack s. Containers are selected from C (line 3), following a specific order to avoid breaking stacking constraints. To avoid having *20'* containers on top of *40'* ones, *20'* containers are selected first. Among containers with the same length, reefer containers are selected before non-reefer since reefer plugs are usually placed at the bottom of the stacks. Finally, when containers have the same length and reefer capabilities heavier containers are selected first. Stowing containers bottom-up in stacks eliminates the possibility of having containers with no support below.

We use the greedy algorithm introduced above to slot plan the vessel following the assignment made by the corresponding master plan. First, we assign the containers from the loadlist I to the locations following the solution from the first phase. Let q_l^τ be the number of containers of type τ assigned to location l during the first phase of the heuristic. For each location $l \in \mathcal{L}$ and type $\tau \in \mathcal{T}$ we select on a first come, first served basis from the loadlist q_l^τ containers of type τ to stow on location l. Since q_l^τ could be a continuous value, unit containers are stowed in the locations with the largest fractional part. Once containers are assigned to locations, we stow independently each $l \in \mathcal{L}$ with the greedy algorithm.

Finally, we use a second run of the greedy algorithm to stow some of the rolled out containers. Let uc_l be the set of containers assigned to location l that could not be stowed by the greedy algorithm, $U = \{\cup_{l \in \mathcal{L}} uc_l\}$ be the set of all containers the greedy algorithm could not stow at each location, and S^V be the set of stacks of the vessel. We change to a vessel perspective by considering all the vessel stacks at once, and use the greedy algorithm to stow the set of containers U in stacks from S^V. The containers that are not stowed at this point are reported as unstowed containers to the stowage coordinator.

5.6 Experiments

In order to evaluate the heuristic introduced in this chapter we use three stowage plans made by stowage coordinators on a vessel with approximately 15.000 TEU capacity. Each stowage plan is a stowage condition of the vessel after visiting the last of six consecutive ports. For our experiments, we extract the planning scenarios at each port used to achieve these stowage conditions and assume that the vessel arrives empty to the first port. The planning scenarios will not include all preferences, as we do not have information about containers sent internally between the ports. However, stowage coordinators have actually turned these planning scenarios into stowage plans, making them suitable for comparing the performance of our heuristic against the stowage coordinators' work. In the few locations where containers with different discharge ports are mixed, something our heuristic does not allow, we assign the discharge port of the majority of containers to the location.

As depicted in Figure 5.2, real-life stowage plans are generated independently for each port in a schedule. Given that our goal is to evaluate the performance of the heuristic introduced in Section 5.5, we generate a data set of 18 instances, i.e., one for each of the six ports in the three real-life stowage plans, and analyse them independently. Table 5.2 specifies the features of the three real-life stowage plans. All the experiments were run on an Ubuntu 10.4 system, with Intel Core 2 Duo, 4 GB of RAM, and CPLEX 12.1.

Table 5.3 presents the results of our experiment. We measure three key factors in our experiments: First, the response time of our heuristic, since one of our goals is to keep the generation of stowage plans within two seconds on average. Second, the number of containers that could not be stowed. Third, the number of cells killed in a stowage plan, which is one of the most common reasons for sub-utilizing the volume capacity of a container vessel.

Our heuristic managed to generate stowage plans for each instance in 1.27 seconds on average, a number well within our desired time for the interactive generation of stowage plans. The longest time was 3.05 seconds (instance 12), while the fastest was 0.17 seconds, (instance 7). There are two important things to notice about the run times presented in Table 5.3. First, it is easy to see that the heuristic spends most of the time solving the LP model. Second, we can observe that there is a direct relation between the

SWP-SC	ID	P	O*20'*	O*40'*	L*20'*	L*40'*	U
	1	1	0	0	182	939	2026
	2	2	182	939	219	750	3779
	3	3	401	1689	563	1439	7220
1	4	4	964	3128	175	412	8219
	5	5	1192	3540	228	685	9817
	6	6	1420	4225	625	747	11936
	7	1	0	0	101	531	1163
	8	2	101	531	141	731	2766
	9	3	242	1262	403	1416	6001
2	10	4	645	2678	407	614	7636
	11	5	1052	3292	199	827	9489
	12	6	1251	4119	827	1141	12598
	13	1	0	0	102	864	1830
	14	2	102	864	242	684	3440
	15	3	344	1548	473	1276	6465
3	16	4	817	2824	116	304	7189
	17	5	933	3128	144	613	8529
	18	6	1077	3741	652	894	10969

Table 5.2: *Instance Overview.* Column SWP-SC shows the identifier of the stowage plan, while ID represents the identifier of the instance, and P the port number in the schedule. O*20'* and O*40'* are the number of *20'* and *40'* release containers on the vessel when it calls the port, respectively. L*20'* and L*40'* represent the number of *20'* and *40'* containers to load when the vessel calls the port, and U the TEU utilization of the vessel when leaving the port.

number of containers to stow and the time used by the heuristic to generate the stowage plans. For each real-life stowage plan, the ports with the greatest number of containers to stow, i.e., ports 3 and 6, are also the ports where the stowage planning takes the longest. It is important to notice as well that the number of release containers does not seem to have a strong influence on runtime, since otherwise, we would see runtimes increasing from port 1 to 6 in all stowage plans.

The maximum percentage of rolled out containers is 1.89% of the containers in the loadlist (instance 6), and 0.58% on average, a considerably low fraction. A difference between the number of *20'* and *40'* containers rolled out can be easily spotted. In 4 out of 18 instances *40'* containers were rolled out, and with respect to the total rolled out containers for all instances, only 16 out of 129 are *40'* long. There are two main reasons for this difference. The first reason is that since there are no odd slots in our test vessel, the number of *20'* containers assigned to a location by the LP must be even. Given that cell capacity must be completely utilized, no cell with an odd number of *20'* containers assigned to it is allowed. Thus, our greedy algorithm will roll out at least one *20'* container when an odd number of *20'* containers has been assigned to a location by the LP. Moreover, when we use our greedy heuristic for the second time (i.e., vessel perspective), to attempt to stow all the rolled out containers, most of the stacks are already full or partially utilized with *40'* containers on top, making it impossible to stow *20'* containers. The second reason relates to the fact that reefer plugs are usually available only at bottom tiers of the vessel locations. In the set of real-life stowage

SWP-SC	ID	P	C	LP(s)	G(s)	D *20'*	D *40'*	KS	KH
	1	1	1121	0.30	0.06	2	0	8	8
	2	2	969	0.74	0.05	3	1	9	10
	3	3	2002	1.86	0.09	1	0	6	5
1	4	4	587	1.01	0.03	1	0	8	5
	5	5	913	1.34	0.04	4	12	12	14
	6	6	1372	2.21	0.06	26	0	7	15
	7	1	632	0.14	0.03	1	0	9	9
	8	2	872	0.57	0.04	3	2	18	14
	9	3	1819	1.47	0.08	3	0	9	16
2	10	4	1021	1.23	0.05	3	0	1	6
	11	5	1026	1.23	0.04	3	1	9	14
	12	6	1968	2.97	0.08	29	0	7	9
	13	1	966	0.23	0.05	0	0	19	23
	14	2	926	0.75	0.06	4	0	11	17
	15	3	1749	1.66	0.10	3	0	6	16
3	16	4	420	0.58	0.03	2	0	8	3
	17	5	757	1.08	0.04	5	0	13	13
	18	6	1546	2.57	0.07	18	0	40	20

Table 5.3: *Experimental Results.* Column SWP-SC presents the identifier of the stowage plan, ID the identifier of the instances, P the port number on the schedule, and C the number of containers to stow. LP(s) and G(s) are the time in seconds used by the LP model and the greedy heuristic, while D*20'* and D*40'* present the number of *20'* and *40'* containers dropped by the heuristic. Finally, KS and KH are the number of cells killed in the plans generated by the stowage coordinators and the placement, respectively.

plans, a considerable number of reefer containers, most of them *20'* long, are loaded in port number six. Since more that 50% of the capacity of the vessel is already utilized at this point, we have already used most of the reefer slots at the bottom tiers of the vessel, even by stowing non-reefer containers. Moreover, most of the few remaining free reefer slots are in stacks partially filled with *40'* containers.

The number of killed cells in our set of instances average 11.1 per instance for the stowage plans made by stowage coordinators, and 12.1 for those made by our heuristic. In 10 out of 18 instances, stowage coordinators managed to kill less cells than our heuristic, while our heuristic killed less cells in 5 instances. Both, stowage coordinators and our heuristic, killed the same number of cells in the remaining three instances. Stowage coordinators killed 20 cells more than our heuristic in the worst case (instance 18), while the heuristic killed 10 more cells than the stowage coordinators, (instance 16). Even though the results of this experiment favours the stowage coordinators, it is important to notice that stowage coordinators have an accumulated expert knowledge of years generating stowage plans, and we are getting close to matching that knowledge (difference on the averaged killed cells is only one), with a heuristic that stows 1,968 containers in a vessel partially full (63%), in three seconds.

It is important to notice when comparing the number of killed cells by the stowage coordinators and our heuristic that we have two simplifications in the generation of stowage plans compared to stowage coordinators: we treat *45'* containers as *40'* and we do not consider lashing constraints in on deck locations. Even with these simplifications,

however, we still think that it is relevant to show how our heuristic performs, compared to stowage coordinators in terms of the number of cells killed, since the impact of considering *45*' containers and lashing constraints will be minimal. Cells are only killed in locations below deck due to the physical limitations imposed by hatch covers. Let us consider what will happen if we allow our heuristic to handle *45*' containers. First, all *45*' containers will be stowed on deck since on deck locations are the only ones with *45*' capacity. This could affect the number of killed cells compared to the ones from the current stowage plans if a *45*' dry cargo container moved on deck is replaced by a *40*' high-cube container. This is, however, very unlikely since most, if not all, *45*' containers are high-cube containers. Any container our heuristic uses to replace the *45*' containers going below deck will be of the same height or smaller and will not increase the number of killed cells.

Roughly speaking, lashing constraints can be seen as constraints that restrict the weight distribution of containers stowed in on deck locations. When the GM increases, these constraints force the number of light containers stowed on deck to be increased. Thus, in cases where all heavy containers are high-cube, we are forced to stow them below deck and an increment in the number of killed cells would be expected. This increment, however, will not be dramatic. In the cost values of the variable in the objective function of our LP model we reward stowing heavy containers in locations below deck, thus, we are generating stowage plans that already stow as many light containers on deck as possible.

To measure the impact of generating master plans with an LP model, thus allowing the assignment of fractional containers into locations, we create a new model for master planning. To do this we add the integrality constraint over the variables representing the number of containers to be stowed in all locations ($x_l^{\tau t}$). We then proceed to generate stowage plans using the new IP model. We generate master plans with 1% and 5% of optimality gap, and evaluate the same criteria as before, i.e., number of rolled out containers, CPU time, and killed cells. Table 5.4 presents the results of using the new IP model for the generation of stowage plans. Since we want to compare the new plans against the ones generated using the LP model, the table presents the difference between the values associated to the stowage plans based on LP master plans and those based on the IP master plans. For instance, in the case of the number of *20*' containers rolled out, a positive value indicates that the stowage plan based on the LP model rolls out more containers than the plan based on the IP model.

From the results in table 5.4 we can see that there is no clear advantage of using an IP model to generate master plans. The improvement on containers rolled out only happens in a few instances and it is negligible, (less than 1%). There are even some cases where the LP based stowage plans roll out less containers, though the number is negligible as well. With respect to the number of killed cells, the IP model actually behaves slightly worse than the LP. On average, stowage plans based on the IP model with a 1% gap killed 0.67 more cells, while the plans based the on IP model with a 5% gap killed 0.83 more cells.

SWP-SC	ID	P	IP(s)		D *20'*		D *40'*		KC	
			1%	5%	1%	5%	1%	5%	1%	5%
	1	1	0.24	0.21	0	0	0	0	1	1
	2	2	1.71	1.56	0	0	0	0	2	2
	3	3	7.68	3.26	0	0	0	0	3	3
1	4	4	138.75	5.67	0	0	0	0	-2	-1
	5	5	6.69	1.40	2	2	0	0	-1	1
	6	6	21.67	3.18	0	6	0	0	-3	-7
	7	1	4.51	4.26	0	0	0	0	-1	-1
	8	2	2.67	2.46	0	0	0	2	-2	-2
	9	3	78.28	49.12	0	0	0	0	2	0
2	10	4	36.16	0.56	0	0	0	0	3	3
	11	5	15.04	1.52	0	0	0	1	-4	-2
	12	6	7.77	6.23	6	2	0	0	-4	-7
	13	1	3.80	3.79	0	0	0	0	-1	-1
	14	2	3.15	3.19	0	0	0	0	0	0
	15	3	48.44	42.79	0	0	0	0	-1	-1
3	16	4	11.32	10.55	0	0	0	0	1	1
	17	5	1.90	1.82	0	0	0	-4	-3	-3
	18	6	11.72	11.15	0	0	0	-4	-2	-2

Table 5.4: *Comparing with Integer Results.* The column SWP-SC represents the identifier of the real-life stowage plan, ID the identifier of the instances, P the port number on the schedule, and the third and fourth columns show the time for the IP model with 1% and 5% gaps, respectively. The next six columns are grouped in pairs and represent the difference of a specific criterion between the results from the LP master plans and the IP master plans with 1% and 5% optimality gap. Columns six and seven present and columns eight and nine show the difference in *20'* and *40'* containers rolled out, respectively. Columns ten and eleven present the difference in number of cells killed.

For six of the instances stowage plans generated based on the LP model killed more cells that those generated based on both IP models. On the other hand, stowage plans based on the LP model killed less cells in eleven of the instances compared to the plans based on the IP model with 1% gap, and in nine of the instances for the IP model with a 5% gap. One of the reasons for this behaviour is related to the objective minimizing the number of killed cells. This objective is based on the premise that in order to avoid the killing of cells the number of HC containers in a location must stay below a precomputed safety limit. Once the number of HC containers in a location goes over this limit, the extra HC are penalized linearly. The relation between HC containers and killed cells, however, is not linear. The slot planning heuristic follows the master plan as closely as possible, thus any inaccuracy in the master plan will impact the slot plan. In the case of the HC containers in IP master plans, the slot planning heuristic attempts to stow the exact number of HC in the locations, even if that number of HC end up killing more cells than anticipated. In the case of LP master plans, the fractional number of HC containers gives the slot planner the flexibility to decide where better to place some of the HC that kill cells.

5.7 Conclusion

This chapter introduced a heuristic that serves as the optimization component of a decision support system to interactively generate container stowage plans. Our heuristic uses the know-how of stowage coordinators represented in a planning scenario to stow containers from a loadlist into vessel slots. Our experiments showed that with 1.27s of runtime on average, our heuristic can be used for interactive optimization. Additionally, the stowage plans produced are competitive with expert users. For a container vessel of approximately 15,000 TEUs, our heuristic managed to only roll out, at most 1.89% of the loadlist and 0.58% on average, and only killed 1 cell more on average than the expert user. As future work we plan to make our stowage plans more accurate by handling *45'* containers and including lashing constraints. We are also interested in studying the impact of reducing the granularity of the weight classes in the LP model's runtime and in the percentage of containers dropped by the greedy heuristic.

Chapter 6

Cargo Composition Problem

In this chapter, we introduce the Cargo Composition Problem (CCP), the problem of evaluating how the stowage characteristics of containers with different features affect performance measures used in the liner shipping domain. To solve the CCP, we introduce the first model with variable displacement that assigns containers to vessel slots and is actually solvable on real instances. Additionally, to increase scalability we have extended the master planning model presented in chapter 4 with the capability of dealing with variable displacement and use it to solve the CCP. This chapter is based on Delgado and Jensen (2013).

6.1 Introduction

Several planning decisions in liner shipping companies depend on the stowage capabilities of container vessels. From strategic decisions involving long term investments (e.g., designing/buying container vessels to fulfill specific demands), to medium term tactical decisions (e.g., vessel deployment selecting the vessels that will cover demands), and daily operational decisions (e.g., selecting containers to transport to increase revenue). What they all have in common is that they often are made under the assumption that a container vessel can be filled up with all kinds of containers, and that these containers can be freely stowed in the vessel. This assumption, however, is far from reality. Thus, having access to an accurate footprint of the stowage capabilities of a container vessel is valuable for liner shipping companies.

In this chapter, we introduce the Cargo Composition Problem (CCP), the problem of evaluating how the stowage characteristics of containers with different features affect important performance measures in liner shipping companies, (e.g., vessel intake and cargo revenue). To solve the CCP, we introduce the first complete model that stows containers into vessel slots that is able to solve real instances. This model can represent variable displacement, a feature of the CCP that has not been considered in previous stowage models. To increase scalability, we present an extension to the master planning model presented in chapter 4 and use it to generate sub-optimal solutions for the CCP.

We evaluated our two models on a benchmark suite with 10 container vessels provided by our industrial collaborator in single and multi-port setups. In the single port setup, we optimized vessel intake and cargo revenue, and stressed both models by optimizing intake of containers with specific features such as weight, length, and height. Our experiments showed that the extended master planning model finds solutions of similar quality to those found by the complete model in a fraction of the time. Moreover, we showed experimentally the advantage of using accurate stowage planning to avoid overestimating the stowage capabilities of a vessel, when doing revenue analysis. In the multi-port setup, we optimized cargo revenue on a set of 360 instances, where the extended master planning model successfully scaled and managed to solve 91.7% of the instances.

The remainder of the chapter is organized as follows. Section 6.2 defines the CCP and section 6.3 presents related work. In section 6.4, we introduce the complete model as an IP that solves the CCP by assigning containers to vessel slots. In section 6.5, we present an extension of the master planning model, introduced in chapter 4, that assigns containers to locations, and we use it to generate sub-optimal solutions for the CCP . Our experiments and results are presented in section 6.6, and conclusions are drawn in section 6.7.

6.2 Problem Definition

The CCP evaluates how the stowage characteristics of containers with different features affect important performance measures used in liner shipping companies, (e.g., vessel intake and cargo revenue). We perform this evaluation by considering only containers matching the features that we want to evaluate and generating seaworthy stowage plans using these containers. During the generation of the stowage plans, the containers to stow are selected according to a gain defined based on their features (i.e, length, height, weight, and reefer capability), and on the performance measure being considered. That is, we compute seaworthy stowage plans that optimize the gain generated by the containers stowed.

When optimizing revenue, additionally to the gain associated with each container, we must also consider costs related to how containers are stowed in the vessel. Container ships in liner shipping companies transport containers between ports on a fixed cyclic route. Containers must be stowed in a vessel such that time at port and fuel consumption are minimized. Time at port is minimized by reducing overstowage, and by efficiently distributing crane moves over the length of the vessel such that the makespan of the quay cranes is reduced. To minimize fuel consumption we reduce the amount of ballast water loaded in the vessel. Ballast water helps fixing stability problems but it increases the displacement of the vessel, forcing the engines to burn more fuel to achieve the desired speed.

All major aspects of stowage must be considered in order for our results to be valuable for the industry. In the CCP, we consider standard ISO containers *20'* and

40' long, with reefer capabilities, and high-cube and normal height. We also consider stacking rules between *20'* and *40'* containers, weight and height stack limits, and we force cells with capacity in both slots to be either empty or with both slots occupied. Additionally, containers are sorted by weight in on deck stacks, a rule of thumb used in the industry to reduce lashing issues. A more accurate modeling of lashing can be done with specialized capacity constraints on stacks and locations. Finally, main stability and stress force calculations are also considered.

In this study we have not considered the line-of-sight constraint and containers with special requirements, (e.g., IMO, pallet-wide, etc). Line-of-sight can be modeled by specialized constraints that reduce on deck stacks capacity according to how the fore draft changes. We do not expect the scalability of our models to be affected by the inclusion of these constraints. With respect to containers with special requirements, they are always placed in specific areas of the vessel, and abstractions of their stowing rules can be modeled with specialized capacity constraints without affecting the overall results produced by the model. Moreover, these containers are underrepresented in the container fleet and a precise modeling of their stowing rules might considerably impact the scalability of the models unnecessarily.

6.3 Literature Review

Previous work on optimization of cargo compositions has been limited to a few publications on revenue management. In Ting and Tzeng (2004), a revenue management model that includes the repositioning of empty containers is introduced. Their stowage model considers volume, *40'*, reefer, and weight capacity for the complete vessel. It disregards, however, the vessel's physical layout and any kind of stability constraints. Likewise, Feng and Chang (2008), introduces a revenue model for short-sea shipping, where total volume and weight capacity of the vessel are the only considerations with respect to stowage of the containers.

Given the similarities between cargo composition optimization and stowage planning, we consider relevant work from this more active field of research that relates to the contributions presented in this chapter. Models that generate stowage plans by assigning containers to vessel slots have mainly been formulated as mathematical programs. One of the most accurate formulations is introduced in Botter and Brinati (1992). This IP model includes several stability and stress force constraints and proposes an approach to perform stability calculations when the displacement of the vessel is variable. The model, however, does not solve in practice. Two more IP formulations are introduced in Ambrosino et al. (2004) and Li et al. (2008), where containers are assigned to slots such that heuristic rules modeling stability constraints are satisfied. Both models consider only standard *20'* and *40'* containers. Ambrosino et al. report that their the IP model do no scale well in practice, while Li et al. test their IP model with random instances on a 800 TEUs vessel, but no evaluation on the scalability of the model is presented.

We refer the reader to sections 4.2 and 5.4 for a review of the academic work on other approaches for stowage planning.

6.4 Complete Model for the CCP

In this section we formulate the CCP as a boolean IP model. This model assigns container types to specific vessel slots satisfying constraints and optimizing objectives described in section (6.2). Recall that a container type τ is a 4-tuple (l, h, r, w), where $l \in L$ is the length of the container in feet, $h \in H$ is the height of the container, $r \in R$ is the reefer properties of the container (reefer or non-reefer), and $w \in W$ is the weight class of the container, expressed as the average weight of the containers in the class in metric tons. We assume that all *20'* container types are of the same height, since *20'* high-cube containers are rare. Below we define decision and auxiliary variables of the IP model. We remind the reader that the sets and constants used in the models presented in this chapter have been previously defined in the section *Sets and Constants*, at the beginning of this document.

Decision variables

$x_{ce}^{t\tau} \in \{0,1\}$	Whether a container of type τ is stowed in slot e of cell c during transport t.
$x_{pu} \in \mathbb{R}^+$	Amount of ballast water in tons loaded at port p in tank u.

Auxiliary variables

$v_{pl}^{\{OH,O\}}$	Hatch (OH), stack (O) overstowing containers at port p in location l.
v_p^C	Crane makespan of port p.
$v_p^W \in \mathbb{R}^+$	Vessel displacement.
$v_{pl}^W \in \mathbb{R}^+$	Weight of containers stowed at port p in location l.
$v_p^{VML} \in \mathbb{R}^+$	Vertical moment of all locations at port p.
$v_p^L \in \mathbb{R}$	Longitudinal center of gravity at displacement interval i and port p.
$v_p^{VM} \in \mathbb{R}^+$	Vertical moment at port p.
$v_p^V \in \mathbb{R}^+$	Vertical center of gravity at port p.
$v_{pi}^W \in \mathbb{R}^+$	Displacement at port p in displacement interval i.
$v_p^M \in \mathbb{R}^+$	Metacenter at port p.
$v_{ps}^B \in \mathbb{R}^+$	Buoyancy force at port p of section between stations s and $s+1$.
$v_{pf}^{S\alpha} \in \mathbb{R}^+$	Shear force at port p fore or aft of frame f.
$v_{pf}^{S\alpha} \in \mathbb{R}^+$	Bending moment at port p fore or aft of frame f .
$\phi_{pc} \in \{0,1\}$	Whether there are containers below cell c going to be discharged at port p.
$\delta_{pl} \in \{0,1\}$	Whether there are containers to be discharged at port p in locations below location l.
$\psi_{pi} \in \{0,1\}$	Indicator variable for displacement interval $i \in I$ at port p.

The objective function of the IP formulation of the cargo composition problem is

$$\begin{aligned} Max : & \sum_{t \in TR} \sum_{c \in \mathcal{C}} \sum_{\tau \in \mathcal{T}} \sum_{e \in \mathcal{E}^\tau} C_t^\tau x_{ce}^{t\tau} \\ & - \sum_{p \in P} (C_p^O (\sum_{l \in \mathcal{L}} v_{pl}^{OH} + \sum_{c \in \mathcal{C}} v_{pc}^O) + C_p^C v_p^C + C_p^U \sum_{u \in \mathcal{U}} x_{pu}) \end{aligned} \tag{6.1}$$

This objective function maximizes a weighted sum of the number of stowed containers based on the individual gain of their type. This gain depends on the kind of analysis being done, (e.g, revenue or intake optimization). The negative impact of hatch and stack overstowage that the stowed containers produce, the crane makespan, and the ballast water necessary to make the vessel seaworthy is also considered. The costs of these latter aspects depend on the scenario the solver is optimizing and will be explained in the experimental section. Next we introduce the stacking constraints that ensure that container stacks in the vessel satisfy stacking rules and capacity limits.

$$\sum_{t \in TR_p^{ON}} \sum_{\tau \in \mathcal{T}} x_{cA}^{t\tau} \leq 1 \quad \forall p \in P,\, c \in \mathcal{C} \tag{6.2}$$

$$\sum_{t \in TR_p^{ON}} \sum_{\tau \in \mathcal{T}^{20}} x_{cF}^{t\tau} \leq 1 \quad \forall p \in P,\, c \in \mathcal{C} \tag{6.3}$$

$$\sum_{t \in TR_p^{ON}} (\sum_{\tau \in \mathcal{T}^{20}} x_{cA}^{t\tau} - x_{cF}^{t\tau}) = 0 \quad \forall p \in P,\, c \in \mathcal{C} \tag{6.4}$$

$$\sum_{t \in TR_p^{ON}} \sum_{c \in \mathcal{C}_{lj}} (\sum_{\tau \in \mathcal{T}} \Pi_{\tau}^{FEU} \sum_{e \in \mathcal{E}^{\tau}} W_{\tau} x_{ce}^{t\tau}) \leq W_{lj}^{40} \quad \forall p \in P,\, l \in \mathcal{L},\, j \in J_l \tag{6.5}$$

$$\sum_{t \in TR_p^{ON}} \sum_{c \in \mathcal{C}_{lj}} \sum_{\tau \in \mathcal{T}^{20}} W_{\tau} x_{ce}^{t\tau} \leq W_{lje}^{20} \quad \forall p \in P,\, l \in \mathcal{L},\, j \in J_l,\, e \in \mathcal{E} \tag{6.6}$$

$$\sum_{t \in TR_p^{ON}} \sum_{c \in C_{lj}} \sum_{\tau \in \mathcal{T}} H_{\tau} x_{cA}^{t\tau} \leq H_{lj} \quad \forall p \in P,\, l \in \mathcal{L},\, j \in J_l \tag{6.7}$$

$$\sum_{t \in TR} \sum_{c \in \mathcal{C}^{\neg R}} \sum_{\tau \in \mathcal{T}^R} \sum_{e \in \mathcal{E}^{\tau}} x_{ce}^{t\tau} = 0 \tag{6.8}$$

$$\sum_{t \in TR} \sum_{c \in \mathcal{C}_{\alpha}^{\neg R}} \sum_{\tau \in \mathcal{T}^R \cap \mathcal{T}^{20}} x_{c\alpha}^{t\tau} = 0 \quad \forall \alpha \in \mathcal{E} \tag{6.9}$$

$$\sum_{t \in TR_p^{ON}} \sum_{\tau \in \mathcal{T}} \sum_{e \in \mathcal{E}^{\tau}} (x_{ce}^{t\tau} - x_{c^+e}^{t\tau}) \geq 0 \quad \forall p \in P,\, l \in \mathcal{L},\, j \in J_l,\, c \in \mathcal{C}_{lj} \tag{6.10}$$

$$\sum_{t \in TR_p^{ON}} \sum_{\tau \in \mathcal{T}} \sum_{e \in \mathcal{E}^{\tau}} (W_{\tau} x_{ce}^{t\tau} - W_{\tau} x_{c^+e}^{t\tau}) \geq 0 \quad \forall p \in P,\, l \in \mathcal{L}^{ON},\, j \in J_l,\, c \in \mathcal{C}_{lj} \tag{6.11}$$

$$\sum_{t \in TR} \sum_{c \in \mathcal{C}^{\neg \alpha}} \sum_{\tau \in \mathcal{T}^{\alpha}} \sum_{e \in \mathcal{E}^{\tau}} x_{ce}^{t\tau} = 0 \quad \forall \alpha \in L \tag{6.12}$$

Constraints (6.2) and (6.3), limit the number of containers to be stowed on the aft and fore slot of all cells at each port p. The former enforces stowing of, at most, a single *20'* or *40'* container in the aft slot. The later forces stowing of, at most, a single *20'* container in the fore slot, since the stowage of a *40'* container is modeled by only occupying the aft slot. Constraint (6.4) ensures that either both slots in a cell hold a *20'* container or none does. Stack weight capacity at each port p for *40'* and *20'* containers is enforced by constraints (6.5) and (6.6), where the former uses the *40'*

weight capacity limit to constrain the combined weight of containers in the stack, while the latter constrains the *20'* capacity in the aft and fore slots of stack i. Containers stowed in a cell either occupy both slots (*40'* containers), or have the same height (*20'* high-cube containers are not considered in our representative problem since they are rare). Thus, considering that in our model it is not possible to stow a single *20'* container in a cell, the aft and fore height will never be out of synchronization. Stack height limits are enforced by constraint (6.7), where only the height of the containers stowed in the aft slots is constrained. Constraint (6.8) removes reefer types from cells with no reefer slots, while constraint (6.9) complements the previous constraint by removing *20'* reefer types from non-reefer slots in cells with a single reefer slot. We do not remove *40'* reefer containers from these slots since they only need a single reefer slot to provide them with electricity. Stacking patterns are enforced by constraints (6.10) and (6.11). The former constraint guarantees that containers have support below, either by stowing them at the bottom of the stack or atop other container(s), and that *20'* containers are not stowed on top of *40'* containers. The latter ensures that containers stowed in stacks on deck are sorted by their weight bottom-up. Slots can be restricted to stow containers of specific lengths, thus, constraint (6.12) removes *20'* and *40'* containers from slots where they cannot be stowed.

Below we define two sets of auxiliary variables used to ease the formulation of stability and stress force requirements.

$$v_{pl}^{W} = \sum_{t \in TR_p^{ON}} \sum_{c \in \mathcal{C}_l} \sum_{\tau \in \mathcal{T}} \sum_{e \in \mathcal{E}^\tau} W_\tau x_{ce}^{t\tau} \quad \forall p \in P, \forall l \in \mathcal{L} \tag{6.13}$$

$$v_{p}^{VML} = \sum_{t \in TR_p^{ON}} \sum_{c \in \mathcal{C}} D_c^V \sum_{\tau \in \mathcal{T}} \sum_{e \in \mathcal{E}^\tau} W_\tau x_{ce}^{t\tau} \quad \forall p \in P \tag{6.14}$$

Auxiliary variables representing the weight of each location are defined by constraint (6.13), while the auxiliary variable representing the vertical moment of all locations at port p is defined by constraint (6.14). Next we present constraints to derive the amount of hatch and stack overstowage and estimate the crane makespan.

$$\sum_{t \in TR_p^{A}} \sum_{c' \in \mathcal{C}_c^-} \sum_{\tau \in \mathcal{T}} \sum_{e \in \mathcal{E}^\tau} x_{c'e}^{t\tau} \leq M_c^S \phi_{pc} \quad \forall p \in P, l \in \mathcal{L}, j \in J_l, c \in \mathcal{C}_{lj} \tag{6.15}$$

$$\sum_{t \in TR_p^{OV}} (\sum_{e \in \mathcal{E}^\tau} \sum_{\tau \in \mathcal{T}} x_{ce}^{t\tau}) + 2(1 - \phi_{pc}) \leq v_c^O \quad \forall p \in P, c \in \mathcal{C} \tag{6.16}$$

$$\sum_{l' \in \mathcal{L}_l^U} \sum_{j \in J_{l'}} (\sum_{t \in TR^A} \sum_{c \in \mathcal{C}_{lj}} \sum_{\tau \in \mathcal{T}} \sum_{e \in \mathcal{E}^\tau} x_{ce}^{t\tau}) \leq M_l^H \delta_{pl} \quad \forall p \in P, l \in \mathcal{L}^{ON} \tag{6.17}$$

$$\sum_{j \in J_l} (\sum_{t \in TR_p^{OV}} \sum_{c \in C_{lj}} \sum_{\tau \in \mathcal{T}} \sum_{e \in \mathcal{E}^\tau} x_{ce}^{t\tau}) - M_l^U (1 - \delta_{pl}) \leq v_{pl}^{OH} \quad \forall p \in P, l \in \mathcal{L}^{ON} \tag{6.18}$$

$$\sum_{t \in TR_p^{A}} \sum_{l \in \mathcal{L}_n} (\sum_{c \in \mathcal{C}_l} \sum_{\tau \in \mathcal{T}} \sum_{e \in \mathcal{E}^\tau} x_{cd}^{t\tau}) \leq v_p^C \quad \forall p \in P, n \in N \tag{6.19}$$

Stack overstowage is calculated by constraints (6.15) and (6.16). Constraint (6.15) sets the indicator variable ϕ_{pc} to be active when there are containers below cell $c \in \mathcal{C}$ to be discharged at port p, while constraint (6.16) defines a lower bound of variable v^O_{pc}. When the indicator variable ϕ_{pc} is active, this bound becomes the number of containers stowed in cell c that are to be discharged in downstream ports, i.e., containers in cell c that are overstowing one or more containers stowed in cells below.

We consider an alternative formulation inspired by cuts (3.16) and (3.17), introduced in section 3.4 to improve the linear relaxation of stack overstowage, where constraint (6.15) is reformulated as the two constraints listed below:

$$\sum_{t \in TR^A_p} \sum_{\tau \in \mathcal{T}} x^{t\tau}_{c'A} \leq \phi_{pc} \quad \forall p \in P,\, l \in \mathcal{L}, j \in J_l, c \in \mathcal{C}_{lj}, c' \in \mathcal{C}^-_c \tag{6.20}$$

$$\sum_{t \in TR^A_p} \sum_{\tau \in \mathcal{T}^{20}} x^{t\tau}_{c'F} \leq \phi_{pc} \quad \forall p \in P,\, l \in \mathcal{L}, j \in J_l, c \in \mathcal{C}_{lj}, c' \in \mathcal{C}^-_c \tag{6.21}$$

This new formulation, however, demands a considerably larger number of constraints. Let us assume n to be the number of cells in a stack i. The current formulation uses $n-1$ constraints to set the indicator variables needed to calculate the stack overstowage in i, while the alternative formulation requires $(n-1)(n)$. Early experimental evaluation of both formulations showed that the alternative formulation did not have a positive impact on the performance of the solver. With respect to hatch overstowage, constraint (6.17) sets the indicator variable δ_{pl} to be active when there are containers being loaded or unloaded at port p, stowed in the locations below location l. Note that the BigM value M^H_l must be twice the capacity of locations below l, since these locations could be emptied and filled up with containers again at the same port p. When the indicator variable δ_{pl} is active, constraint (6.18) defines a lower bound v^{OH}_{pl} of the number of hatch overstowing containers in location l at port p. Finally, we calculate a lower bound for crane utilization by minimizing the makespan of all the cranes that are assigned to the vessel. This calculation is a lower bound since the makespan of the cranes might increase due to constraints limiting operability of the cranes, e.g., cranes cannot work at the same time in adjacent bays. Constraint (6.19) calculates the crane makespan at each port, and defines a lower bound of the continuous auxiliary variable v^C_p.

In the CCP, the displacement of the containers considered in stowage plans used to evaluate a cargo composition is not known in advance. This, combined with the fact that ballast tanks can be freely filled with water makes the vessel displacement variable. An approach to tackle the variability in vessel displacement, introduced by the modeling of ballast tanks, was proposed in section 4.3. In this chapter, we present a generalization of this approach that allows a linear calculation of the center of gravity with an acceptable error, and the use of accurate linear approximations of the hydrostatic data when the variability on vessel displacement depends on cargo and ballast tanks. Let us consider

the calculation of the lcg:

$$\frac{LM^o + \sum_{l \in \mathcal{L}} D_l^L v_l + \sum_{u \in U} D_u^U v_u}{W^o + \sum_{l \in \mathcal{L}} v_l + \sum_{u \in U} v_u} \tag{6.22}$$

This expression is non-linear since two of the terms in the denominator, $\sum_{l \in L} v_l$ and $\sum_{u \in U} v_u$, are variable. We aim at turning the lcg calculation into a linear expression by making the denominator a constant value.

Assume that we can accurately estimate an interval of size W^Δ, that contains the vessel displacement of the desired cargo composition, and that W is the middle point of such an interval. We can transform the lcg calculation into the expression:

$$\frac{LM^o + \sum_{l \in \mathcal{L}} D_l^L v_l + \sum_{u \in U} D_u^U v_u}{W + \Delta_W}, \tag{6.23}$$

where $\Delta_W \in [-\frac{W^\Delta}{2}; \frac{W^\Delta}{2}]$ represents the displacement variation from W. Note that in this lcg calculation the actual vessel displacement must be within $\frac{W^\Delta}{2}$ tons of W (i.e., $W - \frac{W^\Delta}{2} \leq W^o + \sum_{l \in \mathcal{L}} v_l + \sum_{u \in U} v_u \leq W + \frac{W^\Delta}{2}$). Based on this, we can approximate the lcg calculation by removing Δ_W from the denominator of (6.23), turning this non-linear expression into the linear expression:

$$\frac{LM^o + \sum_{l \in \mathcal{L}} D_l^L v_l + \sum_{u \in U} D_u^U v_u}{W}. \tag{6.24}$$

We can reasonably assume that the relative error of this approximation is $\frac{W^\Delta}{2W^{Max}}$, with W^{Max} being the maximum vessel displacement allowed. Note that the same approximation can be used for the vertical and transversal center of gravity.

To complement our approach for addressing variable displacement and for an easier integration with a mathematical model, we propose to define a fixed number of displacement intervals as follows. With W^{Min} being the minimum vessel displacement, we use the value W^Δ to split the displacement range (i.e., $[W^{Min}; W^{Max}]$) into n intervals of size W^Δ, where $n = \lceil \frac{W^{Max} - W^{Min}}{W^\Delta} \rceil$. Additionally, we define $W_i = \frac{W^{Min} + (i-1)W^\Delta + W^{Min} + iW^\Delta}{2}$, for $i \in I$, to be the middle point of displacement interval i.

The assumption that the vessel displacement lies within a given interval is useful for the formulation of accurate linear approximations of the hydrostatic data when the displacement is variable. Figure 4.1 depicts a plot of the hydrostatic data for the trim and metacenter calculation. It is easy to see that planes representing linear approximations of the trim and metacenter calculations are inaccurate in this case. A closer look, however, shows that for a fixed displacement a linear approximation of the hydrostatic data with respect to lcg could be used to approximate the trim and metacenter. Moreover, within small displacement intervals, linear approximations with respect to lcg and vessel displacement can be used to accurately approximate trim and metacenter. Thus, we define linear approximations for each displacement interval $i \in I$ and use them to approximate trim and metacenter when the vessel

displacement is within the corresponding interval. As mentioned in section 4.3, care must be taken since the behaviour of the approximation provided by the linearization is inaccurate for extreme vessel displacements. It is reasonable to assume, however, that the displacement of the cargo compositions will be away from these extremes. For the calculation of the longitudinal buoyancy forces we approximate linearly the hydrostatics of the bonjean at each station point, following the same principle as with trim and metacenter. For a deeper discussion on the buoyancy calculations and the bonjean linearizations we refer the reader to section 4.3.

In chapter 4, we showed through experimental evaluation that the errors introduced by approximating the center of gravity calculations and the linearizations of stability and bonjean data are within an acceptable error. For the generalization presented in this section, it is easy to see that the accuracy of the calculations remains the same. Now we introduce the constraints modeling variable displacement and the calculation of the longitudinal and vertical center of gravity:

$$\sum_{u\in\mathcal{U}} x_{pu} + \sum_{l\in\mathcal{L}} v_{lp}^{W} + W_p^{O} = v_p^{W} \quad \forall p \in P \tag{6.25}$$

$$\sum_{i\in I} W_i^{-}\psi_{pi} \leq v_p^{W} \leq \sum_{i\in I} W_i^{+}\psi_{pi} \quad \forall p \in P \tag{6.26}$$

$$\sum_{i\in I} \psi_{pi} = 1 \quad \forall p \in P \tag{6.27}$$

$$Min_i^{L}\psi_{pi} \leq v_{pi}^{L} \leq Max_i^{L}\psi_{pi} \quad \forall p \in P,\, i \in I \tag{6.28}$$

$$\sum_{l\in\mathcal{L}} D_l^{L} v_{pl}^{W} + \sum_{u\in\mathcal{U}} D_u^{L} x_{pu} + LM_p^{O} = \sum_{i\in I} W_i v_{pi}^{L} \quad \forall p \in P \tag{6.29}$$

$$v_p^{VM} \leq v_p^{V} W_i + (1-\psi_{pi}) Max^{V} \quad \forall p \in P,\, i \in I \tag{6.30}$$

$$v_p^{VM} \geq v_p^{V} W_i - (1-\psi_{pi}) Max^{V} \quad \forall p \in P,\, \forall i \in I \tag{6.31}$$

$$v_{lp}^{VML} + \sum_{u\in\mathcal{U}} D_u^{V} x_{up} + VM_p^{O} = v_p^{VM} \quad \forall p \in P \tag{6.32}$$

$$\sum_{i\in I} v_{pi}^{W} = v_p^{W} \quad \forall p \in P \tag{6.33}$$

$$W_i^{-}\psi_{pi} \leq v_{pi}^{W} \leq W_i^{+}\psi_{pi} \quad \forall p \in P,\, i \in I \tag{6.34}$$

Constraint (6.25) calculates the vessel displacement by considering the weight of the tanks, the cargo loaded in the locations, and the weight of the empty vessel and all constant weights; while constraint (6.26) sets the indicator variable corresponding to the displacement interval that contains the current vessel displacement. Constraint (6.27) restricts the number of active displacement intervals at the same port to be one, and constraints (6.28) and (6.29) approximate the longitudinal center of gravity of the container vessel. Constraint (6.28) sets to zero the variables representing the longitudinal center of gravity of the inactive displacement intervals, while constraint (6.29) calculates the longitudinal center of gravity of the vessel based on the longitudinal moment

of the cargo in the locations, the ballast tanks, the empty vessel including constant weights, and the displacement at each interval. Note that having an auxiliary variable representing the longitudinal center of gravity of each displacement interval is unnecessary to approximate the longitudinal center of gravity of the vessel. These auxiliary variables, however, play an important role in the stability and stress force calculations based on linearizations of hydrostatic data.

The vcg of the vessel is approximated by constraints (6.30)-(6.32). Constraints (6.30) and (6.31) set bounds for the vertical moment at each displacement interval. For the active displacement interval, these two constraints turn into an equality where the vertical moment is equal to the vcg of the vessel multiplied by the representative weight of the active displacement interval. For the displacement intervals that are not active these two constraints have no effect. The vertical moment of the container vessel is calculated by constraint (6.32). Note that the vcg D_u^V of ballast tank u has been approximated to be the vcg of the tank when being 75% full. It can be shown that the integral of the quadratic error for a boxed formed tank is smallest if we use a linearization corresponding to the vcg of the tank being 75% full. Last, constraints (6.33) and (6.34), set the displacement variable corresponding to the active displacement interval to be equal to the vessel displacement, and set the displacement variables of the inactive displacement intervals to zero. Finally, the constraints that restrict stability and stress forces to be within limits are presented below.

$$\sum_{t\in TR_p^{ON}} \sum_{c\in\mathcal{C}} D_c^T \sum_{\tau\in\mathcal{T}} \sum_{e\in\mathcal{E}^\tau} W_\tau x_{cd}^{\tau t} + \sum_{u\in\mathcal{U}} D_u^T x_{pu} + TM_p^O = 0 \quad \forall p \in P \tag{6.35}$$

$$\sum_{i\in I} (A_M^W(W_i) v_{pi}^W + A_M^L(W_i) v_{pi}^L + A_M(W_i)\psi_{pi}) = v_p^M \quad \forall p \in P \tag{6.36}$$

$$\sum_{i\in I} (A_T^W(W_i) v_{pi}^W + A_T^L(W_i) v_{pi}^L + A_T(W_i)\psi_{pi}) = 0 \quad \forall p \in P \tag{6.37}$$

$$v_p^M - v_p^V \geq Min_p^{GM} \quad \forall p \in P \tag{6.38}$$

$$0.5 D_b^B \sum_{s\in\{b,b+1\}} \sum_{i\in I} (A_{Bs}^W(W_i) v_{pi}^W + A_{Bs}^L(W_i) v_{pi}^L + A_{Bs}(W_i)\psi_{pi}) = v_{pn}^B \quad \forall p \in P,\, b \in B \tag{6.39}$$

$$v_{pf}^{S\alpha} = W_f^{S,\alpha} + \sum_{l\in L} G_{lf}^\alpha v_{lp}^W + \sum_{u\in\mathcal{U}} G_{uf}^\alpha x_{up} - \sum_{b\in B} G_{bf}^\alpha v_{pb}^B \quad \forall p \in P,\, f \in F,\, \alpha \in \{A, F\} \tag{6.40}$$

$$v_{pf}^{B\alpha} = W_f^{B,\alpha} + \sum_{l\in\mathcal{L}} A_{lf}^\alpha G_{lf}^\alpha v_{pl}^W + \sum_{u\in\mathcal{U}} A_{uf}^\alpha G_{uf}^\alpha x_{pu} - \sum_{s\in\mathcal{S}} A_{bf}^\alpha G_{bf}^\alpha v_{bp}^B \quad \forall p \in P,\, f \in F,\, \alpha \in \{A, F\} \tag{6.41}$$

$$Min_f^S \leq G_f v_{pf}^{S_{Fore}} + (1 - G_f) v_{pf}^{S_{Aft}} \leq Max_f^S \quad \forall p \in P,\, f \in F \tag{6.42}$$

$$Min_f^B \leq G_f v_{pf}^{B_{Fore}} + (1 - G_f) v_{pf}^{B_{Aft}} \leq Max_f^B \quad \forall p \in P,\, f \in F \tag{6.43}$$

The transversal center of gravity is required to be 0 (heeling angle 0) by constraint (6.35).

Remember that constraints (6.28) and (6.34) force the variables corresponding to vessel displacement, (v_{pi}^{W}), and longitudinal center of gravity, (v_{pi}^{L}), of displacement interval i in port p to be the actual displacement and longitudinal center of gravity of the vessel, respectively, when displacement interval i is active, and zero otherwise. Thus, constraint (6.36) uses variables v_{pi}^{W}, v_{pi}^{L}, and ψ_{pi} to select the linearization coefficients corresponding to the active displacement and approximate the plane representing the metacenter. Likewise, constraint (6.37) approximates the plane representing the trim of the active displacement. This constraint forces the displacement and longitudinal center of gravity of the vessel to match the coordinates in the plane corresponding to trim being equal to zero. The vessel's metacentric high is restricted to be greater than its lower bound by constraint (6.38), and the buoyancy force of the section between two consecutive stations is approximated by constraint (6.39). The approximated plane representing the bonjean at each station is formulated similarly to that of the trim and metacenter. Constraints (6.40)-(6.43) compute shear force and bending moment at each vessel frame. These constraints equivalent to constraints (4.26)-(4.29) presented in section 4.3.1 and will not be discussed any further.

In this section we have introduced the first model with variable displacement, that assigns containers to vessel slots, that is actually solvable in real instances. However, since this model does not scale well in some scenarios, in the next section we present a decomposition that improves scalability with a small impact on model accuracy.

6.5 Decomposed model for the CCP

In this section we present a heuristic decomposition approach for the CCP based on the 2-phase decomposition depicted in Figure 1.3. This decomposition has been used in previous chapters to stow containers into vessel slots. Considering that our interest is not to generate stowage plans but to verify whether cargo compositions that match specific criteria can be generated, we extend the mathematical programming model for master planning introduced in chapter 4, to generate feasible cargo compositions and disregard the slot planning phase. It is important to notice that due to the heuristic nature of the decomposition, cargo compositions generated by optimizing the mathematical model of the master planning phase are not necessarily optimal. Below we introduce the extra constants and variables used in the mathematical programming model.

Decision Variables

$x_l^{t\tau} \in \mathbb{Z}$ Number of containers of type t to be stowed in location l during transport t.

Auxiliary Variables

$v_{pl}^{\{O,OP\}}$ Number of hatch overstowing containers (O), overstowage probability (OP) in location l at port p.

$v_{pl}^{W\{20,40\}}$ Weight of *20'*, *40'* containers stowed in location l at port p.

γ_{pl} Whether there are *20'* containers stowed in location l at port p.

Below we introduce the objective function of the decomposed model used to solve the cargo composition problem.

$$Max: \sum_{t\in TR}\sum_{l\in\mathcal{L}}\sum_{\tau\in\mathcal{T}} C_t^{\tau} x_l^{t\tau} - \sum_{p\in P}(\sum_{l\in\mathcal{L}}(C_p^O v_{pl}^O + C_p^{OP} v_{pl}^{OP}) + C_p^C v_p^C + C_p^U \sum_{u\in\mathcal{U}} x_{up}) \quad (6.44)$$

The objective function is a weighted sum corresponding to that of the IP model. There is, however, a minor difference. Stack overstowage cannot be computed in this model since the level of abstraction of the decision variables disallow it. To address this issue, we replace stack overstowage with a new objective that estimates the probability of *20'* containers being overstowed by *40'* containers within the same location. Though this objective does not measure overstowage within a location as accurately as the stack overstowage does, it provides a good estimate. Next we present the set of location capacity constraints, which are an abstraction of the stacking constraints presented in the IP model. These constraints make it likely to stow the containers assigned to locations in the second phase of the decomposition.

$$\sum_{t\in TR_p^{ON}}\sum_{\tau\in\mathcal{T}^{HC}} x_l^{t\tau} \le K_{pl}^{HC} \quad \forall p\in P,\, l\in\mathcal{L} \quad (6.45)$$

$$\sum_{t\in TR_p^{ON}}\sum_{\tau\in\mathcal{T}} \Phi_\tau x_l^{t\tau} \le K_{pl}^{V} \quad \forall p\in P,\, l\in\mathcal{L} \quad (6.46)$$

$$\sum_{t\in TR_p^{ON}}\sum_{\tau\in\mathcal{T}} \Pi_\tau^{TEU} x_l^{t\tau} \le K_{pl}^{TEU} \quad \forall p\in P,\, l\in\mathcal{L} \quad (6.47)$$

$$\sum_{t\in TR_p^{ON}}\sum_{\tau\in\mathcal{T}^{\alpha}} x_l^{t\tau} \le K_{pl}^{\alpha} \quad \forall p\in P,\, \alpha\in L,\, l\in\mathcal{L} \quad (6.48)$$

$$\sum_{t\in TR_p^{ON}}\sum_{\tau\in\mathcal{T}^{RF}} x_l^{t\tau} \le K_{pl}^{RF} \quad \forall p\in P,\, l\in\mathcal{L} \quad (6.49)$$

$$\sum_{t\in TR_p^{ON}}\sum_{\tau\in\mathcal{T}^{RF}} \Pi_\tau^{TEU} x_l^{t\tau} \le K_{pl}^{RC} \quad \forall p\in P,\, l\in\mathcal{L} \quad (6.50)$$

$$v_{pl}^{W\alpha} = \sum_{t\in TR_p^{ON}}\sum_{\tau\in\mathcal{T}^{\alpha}} W_\tau x_l^{t\tau} \quad \forall p\in P,\, l\in\mathcal{L},\, \alpha\in L \quad (6.51)$$

$$v_{pl}^{W20} \le W_{pl}^{20} \quad \forall p\in P,\, l\in\mathcal{L} \quad (6.52)$$

$$0.5 v_{pl}^{W20} + v_{pl}^{W40} \le W_{pl}^{40} \quad \forall p\in P,\, l\in\mathcal{L} \quad (6.53)$$

Constraint (6.45) restricts the number of high-cubes to be within the capacity of the location. Recall that high-cube containers utilize more volume capacity than dry cargo containers, thus, constraint (6.46) limits the volume of the containers stowed in location l to be within limits. Constraints (6.47)-(6.50) and (6.52)-(6.53) are equivalent to constraints (4.6)-(4.9) and (4.11)-(4.12) presented in section 4.3.1, respectively, and will not be discussed any further. Constraint (6.51) defines auxiliary variables v_{pl}^{W20} and v_{pl}^{W40}. Though this constraint resembles constraint (4.10), they differ with respect

to the coefficients of the variables. In the model introduced in section 4.3.1 the weight of a container type changes according to the transports, while in the model presented in this section, container types have fixed weight. We now present two constraints to calculate auxiliary variables used in the stability and stress force computations.

$$v_{pl}^{W} = v_{pl}^{W^{20}} + v_{pl}^{W^{40}} \quad \forall p \in P,\, l \in \mathcal{L} \tag{6.54}$$

$$v_{p}^{VML} = \sum_{t \in TR^{ON}} \sum_{l \in \mathcal{L}} D_{pl}^{V} \sum_{\tau \in \mathcal{T}} W_{\tau} x_{l}^{t\tau} \tag{6.55}$$

Constraint (6.54) calculates the weight of the cargo stowed in a location at a specific port, while auxiliary variable v_p^{VML} (the vertical moment of the cargo) is calculated by constraint (6.55). Constraints on the stability and stress forces of the vessel are listed below.

$$\sum_{u \in \mathcal{U}} D_{u}^{T} x_{pu} + TM_{p}^{0} = 0 \quad \forall p \in P \tag{6.56}$$

$$(6.25) - (6.34)$$

$$(6.36) - (6.43)$$

Constraint (6.56) calculates the transversal center of gravity of the vessel. Notice that locations are not included in this calculation since due to their construction, their tcg is assumed to be in the center of the vessel. Figure 2.5 shows the locations of a bay. For the locations in the center, we can assume that during the slot planning phase weights will be evenly distributed within the location, displacing its tcg towards the center of the vessel. With respect to the outer locations, we can see in Figure 2.5 that storage areas on the starboard and port side form a single location. Since these two storage areas are usually symmetric, we also assume an even distribution of the weights during slot planning. Thus, the tcg of the outer locations is displaced towards the center of the vessel as well. The remaining stability and stress force calculations for this model are the same as for the IP model. To derive the objective values we introduce the constraints below:

$$\sum_{k \in \mathcal{L}_{l}^{U}} \Big(\sum_{t \in TR^{A}} \sum_{\tau \in \mathcal{T}} x_{l}^{t\tau} \Big) \leq M_{l}^{U} \delta_{pl} \quad \forall p \in P,\, l \in \mathcal{L}^{ON} \tag{6.57}$$

$$\sum_{t \in TR_{p}^{OV}} \sum_{\tau \in \mathcal{T}} x_{l}^{t\tau} - M_{l}^{H}(1 - \delta_{pl}) \leq v_{pl}^{O} \quad \forall p \in P,\, l \in \mathcal{L}^{ON} \tag{6.58}$$

$$\sum_{t \in TR_{p}^{A}} \sum_{l \in \mathcal{L}_{n}} \sum_{\tau \in \mathcal{T}} x_{l}^{t\tau} \leq v_{p}^{C} \quad \forall p \in P,\, n \in N \tag{6.59}$$

$$\sum_{t \in TR_{p}^{A}} \sum_{\tau \in \mathcal{T}^{20}} x_{l}^{t\tau} \leq M_{l}^{H} \gamma_{pl} \quad \forall p \in P,\, l \in \mathcal{L} \tag{6.60}$$

$$\sum_{t \in TR_{p}^{OV}} \sum_{\tau \in \mathcal{T}^{40}} x_{l}^{t\tau} - 0.5 M_{l}^{H}(1 - \gamma_{pl}) \leq v_{pl}^{OP} \quad \forall p \in P,\, l \in \mathcal{L} \tag{6.61}$$

Constraints (6.57) and (6.58) model hatch overstowage similar to the IP model, where indicator variables δ_{pl} is activated when there are containers being loaded or unloaded at port p in the locations below l. Then, constraint (6.58) uses the indicator variables to calculate a lower bound for hatch overstowage. Also, similar to the computations in the IP model, constraint (6.59) estimates the crane makespan. Finally, since *20'* containers have to be stowed under *40'* containers within a location, when there are *40'* containers stowed in location l to be discharged after port p there is a probability that these containers overstow *20'* containers to be loaded or discharged at port p in location l. Constraint (6.60) and (6.61) model the probability of overstowage within a location. Indicator variable ϕ_{pl} is activated by constraint (6.60) when there are *20'* containers stowed in l to be loaded or discharged at port p. Constraint (6.61) then provides a lower bound for the number of *40'* containers being discharged after p, that will potentially overstow *20'* containers to be loaded or discharged at port p.

6.6 Experiments

We test the two models presented in this chapter on a benchmark suite of ten container vessels provided by our industrial collaborator. Table 6.5 presents a summary of the vessels, sorted by TEU capacity. These vessels represent most of the main vessel categories with respect to TEU capacity, including Feeder, Panamax, Post Panamax, and Ultra Large container vessels. Note that the TEU capacity only reflects the vessel size for the IP model introduced in section 6.4. In the model introduced in section 6.5, containers are assigned to vessel locations instead of stowing them in vessel slots, thus, a more accurate measure of vessel size for this model is the number of vessel locations.

ID	Capacity (TEU)	Locations
1	15000	92
2	10000	87
3	10000	100
4	9000	100
5	8000	80
6	7000	90
7	7000	65
8	5000	71
9	4000	61
10	3000	40

Table 6.5: *Vessel features.* Container vessels features. The first column represents the vessel ID. The second column shows the TEU capacity of the vessels rounded to the nearest thousand due to confidentiality issues, while the third column provides the number of vessel locations.

Recall that the decision variables $x_l^{t\tau}$ of the decomposition model presented in section 6.5 represent the number of containers of type τ assigned to location l during transport t. We evaluate experimentally the impact on CPU time and solution quality when the integrality constraint on variables $x_l^{t\tau}$ is removed. The result of this relaxation is that the solutions produced by the decomposition will include continuous numbers of containers to be stowed in locations. Though it is physically impossible to do so, the impact of fractionality is eased by the fact that the TEU capacity of the locations (from 62 to up to 171 TEUs in general, though a few locations can be smaller), allows rounding to be performed without having a significant impact, (Pacino et al. (2011)). For the remainder of the chapter, we denote the decomposition model with integrality constraints as the *IP decomposition*, and the model where these constraints are removed as the *MIP decomposition*.

We have devised two main sets of experiments to evaluate the performance of the approaches. In the first set, we focus on evaluating the behaviour of the models in a single port scenario, where we stress the models by optimizing TEU intake, intake of containers with specific features (e.g., reefer, high-cube, *20'*, *40'*, and different weight classes), while requiring maximum intake, and cargo revenue. In the second set, we define a multi-port scenario to analyze the impact of different features (e.g., number of discharge ports, transport density, and number of types considered), on the complexity of finding cargo compositions that optimize revenue. In both sets of experiments we split the displacement range in intervals of size 1% of the maximum displacement to keep the approximations of the center of gravity and the linearization of hydrostatic data with an acceptable error. We constrain the displacement to be greater than 50% of maximum displacement, a reasonable assumption given the type of experiments performed in this section, forcing our models to only consider a total of 50 displacement intervals. Also note that for some of the vessels from our benchmark suite, the linearization of the hydrostatic data was problematic with displacements below 50%. All the experiments were run on as AMD Opteron processor at 2.8 Ghz, with 4GB of memory. The models have been implemented using the modeling language OPL and executed using CPLEX 12.5. The time limit for all the experiments was 3,600s.

6.6.1 Single Port Experiments

In this section we analyse the CPU time and solution quality of the results produced by the IP model, the IP decomposition, and the MIP decomposition mentioned above in a single port setup. For these experiments, the objective function of each model is limited to consider the stowed containers and their related coefficients, disregarding overstowage, crane split, and ballast water penalization.

ID	$\mathcal{T}^{32}$			$\mathcal{T}^{20}$			$\mathcal{T}^{40}$		
	IP(%)	DMIP(%)	DIP(%)	IP(%)	DMIP(%)	DIP(%)	IP(%)	DMIP(%)	DIP(%)
1	100.00	100.00	99.99	90.40	90.40	90.39	100.00	100.00	100.00
2	100.00	100.00	100.00	85.90	85.90	85.90	99.62	99.62	99.62
3	99.98	100.00	99.98	85.59	85.59	85.59	99.98	100.00	99.98
4	96.73	96.75	96.73	78.12	78.12	78.12	96.73	96.75	96.73
5	100.00	100.00	100.00	97.19	97.19	97.19	100.00	100.00	100.00
6	96.03	96.03	96.03	77.07	77.07	77.07	95.82	95.82	95.82
7	100.00	100.00	100.00	96.22	96.22	96.22	99.82	99.82	99.82
8	100.00	100.00	100.00	92.79	92.79	92.79	100.00	100.00	100.00
9	100.00	100.00	100.00	95.58	95.58	95.58	96.53	96.53	96.53
10	100.00	100.00	100.00	89.14	89.14	89.14	100.00	100.00	100.00

Table 6.6: *Intake optimization.* First column is the ID of the vessel. Next, each set of three consecutive columns is the percentage of vessel capacity that could be stowed in the vessels (maximum intake), considering only container types of the sets $\mathcal{T}^{32}$, $\mathcal{T}^{20}$, $\mathcal{T}^{40}$, by the IP (IP) model, the MIP decomposition (DMIP), and the IP decomposition (DIP), respectively.

Intake optimization

The first analysis we perform is intake optimization. We attempt to find a cargo composition that maximizes intake for each container vessel, selecting containers from three different type-sets. These type-sets mainly differ from each other in two ways, their size and the length feature of the types they consider. Thus, since the size of the type-sets is related to the number of variables of the models and a considerable number of constraints depend on the length of the container types, in this experiment we study the impact of these two features in CPU time and quality of the solutions generated by our models. Remember that a type τ is a 4-tuple (l, h, r, w). For the first type-set, $\mathcal{T}^{32}$, we define $l \in L = \{20, 40\}$, $h \in H = \{DC, HC\}$ (dry cargo or high cube), $r \in R = \{RF, NR\}$, $w \in W = \{3, 6, 9, 14, 21, 27\}$, and formally restrict $\mathcal{T}^{32}$ to be: $\mathcal{T}^{32} = \{(20, DC, r, w) \,|\, r \in R, w \in W \setminus \{6\}\} \cup \{(40, h, RF, w) \,|\, h \in H, w \in W \setminus \{6\}\} \cup \{(40, h, NR, w) \,|\, h \in H, w \in W\}$. For the second set, $\mathcal{T}^{20}$, we redefine $l \in L = \{20\}$, $h \in H = \{DC\}$, $r \in R = \{NR\}$, $w \in W = \{3, 9, 14, 21, 27\}$, and for the third set, $\mathcal{T}^{40}$, we only redefine $l \in L = \{40\}$ and $w \in W = \{3, 6, 9, 14, 21, 27\}$ with respect to the values in the second set. Table 6.6 presents the optimal intake for the three different type-sets defined above, solved by the complete model (IP), the IP decomposition (DIP), and the MIP decomposition (DMIP).

For each type-set, the three approaches were almost able to stow the same number of containers for all vessels, with a maximum variation of 0.02% (type-sets $\mathcal{T}^{40}$ and $\mathcal{T}^{32}$, vessels 3 and 4, and type-sets $\mathcal{T}^{32}$ and $\mathcal{T}^{20}$, vessel 1), showing little impact of using the IP and MIP decompositions for solving the cargo composition problem in this case. With respect to the impact on maximum intake when considering type-sets that include different lengths, we take as a reference point the maximum intake reached with type-set $\mathcal{T}^{32}$ for each vessel, since this type-set includes types with both lengths, (*20'* and *40'*). The most drastic intake reduction comes from the type-set with only *20'*

containers ($\mathcal{T}^{20}$), with loses starting at 2.81% (vessel 5), and going all the way down to 18.96% (vessel 6), with respect to $\mathcal{T}^{32}$ intake. There are two reasons for this intake loss. The first is that there are sections of the vessel where it is physically impossible to stow *20'* containers, and without *40'* containers available, these sections remain empty. The second reason is the *20'* weight capacity. This capacity is usually smaller than the *40'* weight capacity on deck, further reducing the number of *20'* containers that can be loaded in the container vessels. For the third type-set, $\mathcal{T}^{40}$, none of the three models was able to reach the $\mathcal{T}^{32}$ intake for four of the vessels (vessels 2, 6, 7, and 9), though the capacity loss, ranging from 0.18% (vessel 7), to 3.47% (vessel 9), was less than that of type-set $\mathcal{T}^{20}$. The main reason for the reduction in this case is the fact that there are also some vessel sections where it is impossible to stow *40'* containers (e.g., odd slots).

In conclusion, the length of the types included in the type-set has a considerable impact on the maximum intake achieved for a vessel. In our results, vessels are more suitable for transporting *40'* than *20'* containers, since we can observe a reduction in maximum intake in all the vessels when using type-set $\mathcal{T}^{20}$, while only in four (vessels 2, 6, 7, and 9), when using type-set $\mathcal{T}^{40}$. Moreover, the difference in the reductions is considerable, with a maximum reduction of 18.96% for type-set $\mathcal{T}^{20'}$, and 3.47% for type-set $\mathcal{T}^{40}$. This impact, however, depends on the vessel's physical layout.

It took just a few seconds for the IP and MIP decompositions to maximize intake for all the vessels when considering all three type-sets , with the MIP decomposition finding optimal solutions for all instances in less than a second. The left plot in Figure 6.1 depicts the CPU time of the IP decomposition, indicating a tendency towards the type-set $\mathcal{T}^{32}$ being the hardest to optimize. It took at most 12.03s (vessel 1), and 4.03s on average for the IP decomposition to find the maximum intake for type-set $\mathcal{T}^{32}$, while it took 0.705s and 0.6s in average to find the maximum intake for type-sets $\mathcal{T}^{20}$ and $\mathcal{T}^{40}$, respectively.

With respect to the IP model, the right plot in Figure 6.1 depicts the time it took to find the maximum intake for each vessel and each type-set. A similar tendency to that of the IP decomposition is observed, where instances considering type-set $\mathcal{T}^{32}$ are the hardest to optimize. In this case, when optimizing type-set $\mathcal{T}^{32}$ vessel 2 took the longest time (1,052.01s), and the average was 228.26s. Optimizing intake when considering type-sets $\mathcal{T}^{20}$ and $\mathcal{T}^{40}$ took 98.3s and 32.55s on average, respectively. Thus, from the two plots depicted in Figure 6.1 we can conclude that there is a tendency towards larger type-sets taking more CPU time to find optimal solutions.

Type intake optimization

Next, we attempt to stress our models by optimizing specific container types using type-set $\mathcal{T}^{32}$. We constrain the intake in this experiment to be within one percent of the maximum intake from table 6.6. We start by optimizing intake of container types with specific length, height, and reefer capabilities. Table 6.7 presents the results for

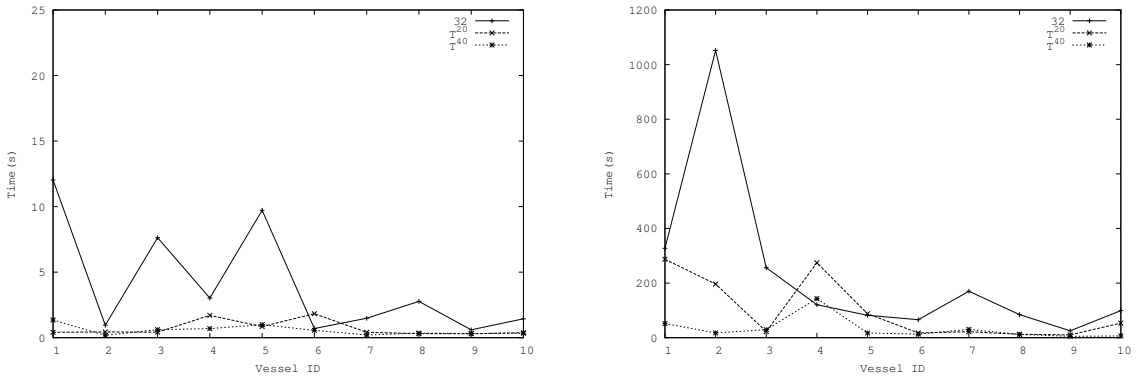


Figure 6.1: CPU time of intake optimization of the, (a) IP decomposition. (b) IP model.

reefer and high-cube containers, while table 6.8 provides the results for *20'*, and *40'* containers.

ID	R			HC		
	IP(%)	DMIP(%)	DIP(%)	IP(%)	DMIP(%)	DIP(%)
1	100	100	100	-	85.32	-
2	-	100	100	-	87.87	-
3	100	100	-	69.52	70.24	-
4	-	100	100	66.40	67.15	67.12
5	-	100	100	72.04	74.62	-
6	100	100	100	-	64.40	64.30
7	100	100	100	-	72.24	72.11
8	100	100	100	74.31	78.00	77.76
9	100	100	100	83.22	96.53	96.53
10	100	100	100	80.11	85.86	85.30

Table 6.7: *Type intake optimization for reefer and high-cube containers.* First column is the ID of the vessel. The first set of three consecutive columns (second, third and fourth columns) are the percentages of reefer slot capacity loaded with reefer containers, while the second set of three consecutive columns are the percentage of TEU capacity loaded with high-cube (HC) containers by the IP (IP) model, the MIP decomposition (DMIP), and the IP decomposition (DIP), respectively, while keeping intake within 1% of the optimal. The dash represents instances that could not be solved within the 3,600s time limit.

Given the different levels of accuracy between the IP model and the decompositions, we expect optimal solutions to be slightly different when we stress the models. As seen in tables 6.7 and 6.8, most of the discrepancies involving a considerable number of containers took place when optimizing high-cubes. In all solutions found by the IP model in this case, less high-cube containers were stowed than in their decomposition counterparts. One reason for this is the abstraction of stack capacities into location capacity made in the decompositions and the accumulated error that this abstraction generates. The volume capacity of a stack fits a specific mix of high-cube and normal contain-

ID	20'			40'		
	IP(%)	DMIP(%)	DIP(%)	IP(%)	DMIP(%)	DIP(%)
1	90.40	90.40	90.39	100.00	100.00	100.00
2	85.90	85.90	85.90	99.62	99.62	99.62
3	85.59	85.59	85.59	99.98	99.99	99.98
4	-	78.12	78.12	96.73	96.74	96.73
5	97.19	97.19	97.19	100.00	100.00	100.00
6	77.07	77.07	77.07	95.82	95.82	95.82
7	96.22	96.22	96.22	99.82	99.82	99.82
8	92.79	92.79	92.79	100.00	100.00	100.00
9	95.58	95.58	95.58	83.22	96.53	96.53
10	89.14	89.14	89.14	100.00	100.00	100.00

Table 6.8: *Type intake optimization for 20' and 40' containers.* First column represents the ID of the vessel. The first set of three consecutive columns (second, third and fourth columns), presents the percentage of 20' (TF), and 40' (FF) containers loaded by the IP (IP) model, the MIP decomposition (DMIP), and the IP decomposition (DIP), respectively, while keeping intake within 1% of the optimal. The dash represents instances that could not be solved within the 3,600s time limit.

ers. Stowing mixes with more normal or high-cube containers force a sub-utilization of the stack's volume capacity, generating a residual capacity where no container can be stowed. When abstracting volume stack capacities into volume location capacity, the residual capacity of the stacks is aggregated such that more high-cube containers can be assigned to a location than actually can be stowed in the stacks. Thus, when optimizing high-cube containers the decomposition erroneously uses the aggregated capacity to stow containers that the IP model will not.

Another discrepancy between optimal values generated by the IP model and the decompositions occurs when optimizing *40'* containers intake for vessel 9 (13.31%). We can see from table 6.6 that when optimizing intake with type-set $\mathcal{T}^{40}$ the results are the same for the IP model and the decompositions. Since $\mathcal{T}^{40} \subset \mathcal{T}^{32}$, we would expect the similar results in table 6.8. As we can see, however, the constraint forcing intake to be within 1% of the optimal impacts the number of *40'* containers that can be stowed in vessel 9. To achieve optimal intake in this specific case, the IP model must load more *20'* containers than the decompositions, indicating an imprecision in the abstraction of stacking constraints related to the interaction of *20'* and *40'* containers.

With respect to CPU time and complexity of finding optimal solutions, the MIP decomposition easily finds optimal solutions for all vessels in 0.43 seconds, on average, and a maximum time of 0.51 seconds. The IP decomposition managed to easily find solutions optimizing *20'* and *40'* containers, but struggled with reefer and high-cube containers. Optimizing high-cube containers was the hardest, finding solutions only for six out of ten vessels. Among the vessels that were solved, no optimal solution was found for four of them (vessels 6-8, and 10), though their final gap was less that 0.7%. With respect to the cases where reefer containers were optimized, there was only one

vessel for which no solution was found, while optimal solutions were found for the rest.

The IP model also struggled when optimizing high-cube and reefer containers, and slightly when optimizing *20'* and *40'* containers. With respect to the optimization of *40'* containers, no optimal solution was found for vessel 9, though the final gap was less than 0.1%. In the case of the *20'* container optimization, optimal solutions were found within the time limit for all the vessels solved. For two of the vessels for which solutions were found when optimizing high-cube containers, optimal solutions could not be generated within the time limit (vessels 4 and 9), though their gap was less than 0.2%. When optimizing reefer containers, optimal solutions were found for the six vessels solved.

There are two important remarks to be made. First, we can see that in some specific cases (high-cube optimization, optimizing *40'* for vessel 9), there is an impact of using the decomposition model on the accuracy of the solutions. Though the inaccuracies can be considerable (up to 13%), they can be addressed by adding a more refined set of capacity constraints into the decomposition model. For instance, in the case of the high-cube optimization, the volume capacity limits could be modified based on the combination of high-cube and normal containers stowed in the location, instead of being fixed as they are in the current model. Second, the MIP and IP decompositions generate almost identical solutions in all cases, with the MIP decomposition using a small fraction of the CPU time used by the IP decomposition.

Now we proceed to maximize the loading of containers with different weights classes. Table 6.9 presents the results of optimizing containers with specific weights from the type-set $\mathcal{T}^{32}$. As with the previous experiments, we constrain the intake to be within

ID	W9			W21			W27		
	IP	DMIP	DIP	IP	DMIP	DIP	IP	DMIP	DIP
1	12759	12838.97	12838	5093	5147.15	5147	3860	3860.36	3860
2	8612	8541.15	8540	4854	4551.63	4547	3751	3694.39	3693
3	7670	7365.58	7361	4929	4587.53	-	3871	3805.65	3800
4	3320	7341.29	7337	4570	4363.19	-	3506	3489.69	3489
5	8217	8217.00	8217	4836	4835.09	4834	3694	3706.77	-
6	6206	6218.86	6216	-	3856.97	-	2957	2959.43	-
7	6292	6202.45	6202	3663	3599.03	3596	2831	2795.60	2795
8	4352	4078.44	4078	2788	2612.54	2606	2319	2178.49	2167
9	3578	3465.54	3465	1996	1903.63	1901	1635	1582.25	1577
10	2282	2253.93	2253	1375	1276.78	1276	1062	1054.81	1054

Table 6.9: *Weight class optimizations.* First column is the ID of the vessel. Next, each set of three consecutive columns are the maximum number of containers weighing 9 (W9), 21 (W21), and 27 (W27) tons stowed by the IP (IP) model, the MIP decomposition (DMIP), and the IP decomposition (DIP), respectively, while reaching intake within 1% of the optimal. The dash represents instances that could not be solved within the 3,600s time limit

1% of the one from table 6.6. At first glance, we can easily see from table 6.9 that as we increase the weight of the containers to optimize, their intake is reduced. Time

wise, the MIP decomposition managed to compute the optimizations within, at most, 3 seconds. The IP decomposition managed to optimize containers weighing 9 tons within at most 75 seconds, but struggled with optimizing containers weighing 21 and 27 tons. When optimizing 21 ton containers, the IP decomposition did not manage to find an optimal solution for vessel 5, though the remaining six vessels for which a solution was found were solved optimally within 7 seconds, including the two largest vessels. The three vessels that were not solved in this case are among the ones with the largest number of locations. When optimizing 27 ton containers, the IP decomposition found optimal solutions for seven of the solved vessels within 9 seconds, while the solution for vessel 10 was found in 119 seconds. The IP decomposition approach does not seem to be affected by the vessel size when optimizing weight classes, though there is clear tendency of the problem becoming harder as container weight increases.

As with the IP decomposition, the IP model also becomes harder to solve when we increase the container weight we want to optimize. When optimizing 9 tons containers, we found optimal solutions for all but one vessel (vessel 4, 55% gap), in an average time of 1,011.14s. For the optimization of containers weighing 21 and 27 tons, the number of optimal solutions kept decreasing, with the optimization of 27 ton containers being the most affected, (only the last three vessels solved optimally). The average time for optimizing 21 ton containers was 1,191.88s, and 2,526s for optimizing 27 tons. In the case of the IP model there also seems to be a tendency that relates vessel size with CPU time, since the vessels that cannot be solved optimally are usually the largest ones.

In table 6.9 we can also see differences between optimal solutions produced by the IP model and the decompositions. In two of these situations, when optimizing 9 ton containers for vessel 4 and when optimizing 21 ton containers for the vessel 1, there was an optimality gap in the IP results of 55% and 1%, respectively, explaining the difference in the results. In the cases where the difference favours the IP model, such as when optimizing 21 and 27 ton containers for most of the vessels, this difference can be explained by how the vcg of the vessel is calculated in the IP model and the decompositions. Recall that for the decompositions we pre-compute a fixed vcg for each location under the assumption that the weight of the containers stowed in them will be uniformly distributed. The IP model allows a more accurate computation of the vcg in the locations, since it can use the exact position of the containers to do so. In the cases where containers of the same weight are stowed in a location this does not have any impact. However, when containers of different weights are stowed in a location the vcg's location can be lowered by stowing heavy containers at the bottom cells and light containers at the top. This reduction allows the stowing of heavier containers than those stowed when the vcg is considered fixed, as in the decompositions.

In conclusion, the MIP decomposition scales well when optimizing containers with specific features, solving all the instances in less than 5s and producing solutions with negligible differences with respect to those produced by IP decomposition. The IP model and IP decomposition, however, have problems coping when the models were stressed with these kind of optimizations, leaving several instances unsolved within the time limit. We also observed small inaccuracies between the intakes in the solutions

produced by the decompositions and the IP models. Though there was a case where the discrepancies reached 13.31% (vessel 9, *40'* intake optimization), in the rest of the cases they did not go over 5%. Moreover, these discrepancies can properly be tackled with a more refined set of capacity constraints and a more accurate approximation of the vcg of the locations.

Revenue optimization

Now we introduce a simplified revenue model for loading containers in a single port to evaluate the impact of the accuracy of the stowage models when doing revenue analysis. We assume that the revenue of a dry *40'* container is US$1,000. Based on interviews with our industrial collaborators and brokers, we calculate the revenue for containers with other features as follows: *20'* containers generate 62,5% of the revenue of a *40'* container. Reefer containers are 75% more profitable than normal containers and high-cube containers are US$100 more profitable than normal containers. *20'* containers weighing between 18 and 21 tons are US$150 more profitable than lighter containers, while those weighing between 21 and 24 tons are US$350 more profitable. Containers weighing more than 24 tons are US$500 more profitable. Finally, we assign a profit of US$100 for empty containers. Though these containers do not generate a profit for the liner shipping company, they need to be constantly repositioned due to trade imbalance between commercial zones. We assign them a small profit margin so that they are loaded in the vessel after more profitable containers have been loaded.

Figure 6.2 depicts the results of optimizing revenue for a single port based on the container revenue model described above for the IP model and the IP and MIP decompositions. With respect to the revenue reported for each model, there is a small difference between the decompositions and the IP model, favouring the IP model for vessels 2, 3, 4, and 8. As we mentioned before, the IP model has a more accurate definition of the vcg of the locations than that of the decompositions. This accuracy allows the IP model to generate stowage plans loading heavier containers, the most profitable containers. Time-wise, the IP and MIP decompositions find optimal solutions in at most 98.1s and 0.31s, respectively, though in the case of the IP decomposition the second largest time is 9.33s. With respect to the IP model, no optimal solution was found for the first three vessels (though the gap was less than 0.03%), and the optimal solution for the smallest vessel was found in 22.86s. For this model, it is clear that the vessel size impacts the time it takes to find optimal solutions.

To test the impact of modeling stowage accurately, we have implemented a simplified model that only constrains the TEU capacity of the vessel and the cargo weight to be within limits, similar to the model described in Feng and Chang (2008), and compare the results with the models introduced in this chapter. Figure 6.2 depicts the results of this experiment. There is a clear overestimation of the revenue of all vessels. On average, the revenue is overestimated by 7.84%, going as low as 4%, and reaching 11.43% in the worst case.

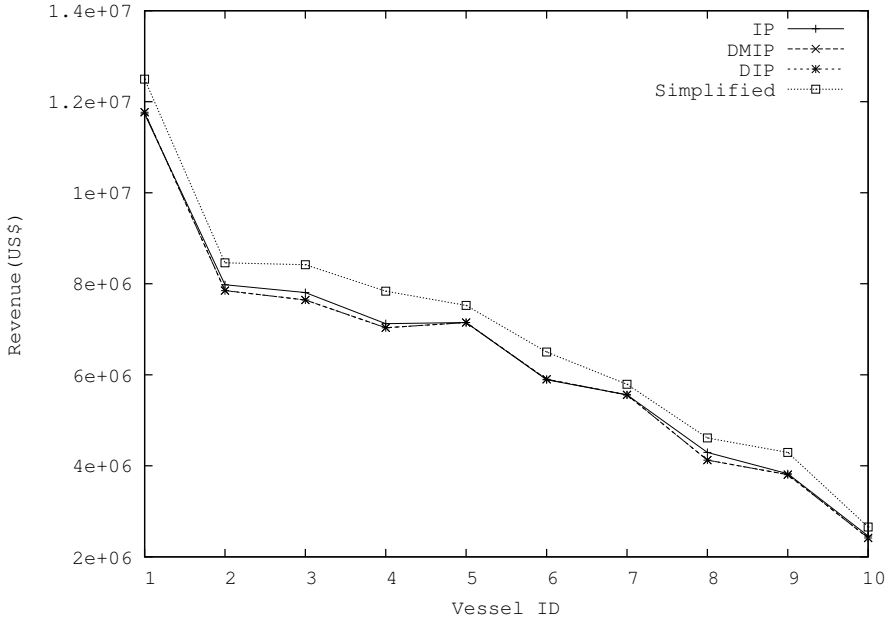


Figure 6.2: Revenue model for single port with accurate stowage planning models and a simplified model.

Variable displacement relaxation

The models introduced in this chapter used an extra set of auxiliary boolean variables to model variable displacement. As mentioned above, for this experimental setup we have used 50 intervals, thus introducing 50 auxiliary boolean variables. In this section, we study the behaviour of these variables in the linear relaxation of our models. For each model, we solve 170 instances optimizing intake, type intake, and revenue, and determine the number of boolean variables representing displacement intervals that are greater than zero. Note that we only consider a single decomposition model since the linear relaxation of the IP and MIP decompositions are the same. Table 6.10 presents the results of the IP model and the decomposition.

As shown in table 6.10 that a considerable part of the instances have two boolean variables or less that are different from zero in the linear relaxation, (87.06% and 78.82% for the IP model and the decomposition, respectively). Moreover, there are only 4 instances out of 340 with more than three boolean variables different to zero. Additionally, the linear relaxation seems to be tighter in the IP model than in the decomposition since in 18.24% of the IP model instances the solution was integer, while this was the case only for 7.65% of the instances solved by the decomposition. Thus, our experiments show a strong linear relaxation of the model with respect to the auxiliary boolean variables representing the displacement intervals, reducing the overhead of branching on these variables.

Variables	IP		D	
	Instances	Percentage	Instances	Percentage
1	31	18.24	13	7.65
2	117	68.82	121	71.18
3	21	12.36	33	19.41
4	1	0.59	2	1.18
5	0	0.00	1	0.59

Table 6.10: *Linear relaxation of auxiliary boolean variables representing the displacement intervals.* First column is the number of boolean variables greater than 0; second and third columns are the number and percentage of instances for the IP model; and fourth and fifth columns the number and percentage of instances for the decomposition.

6.6.2 Multi-Port Experiments

In this section we evaluate the IP model and the IP and MIP decompositions in a multi-port setup. We study the impact that the number of ports in the schedule, the size of the type-set, and the transport density have on the revenue, the response time, and the integrality gap (if no optimal solution is found within the time limit). We define transport density as the percentage of selected transports leaving a port. For instance, in a schedule with three ports, the total number of transports leaving the first port is two, e.g., $\langle 1, 2\rangle$ and $\langle 1, 3\rangle$. A transport density of 100% includes both transports, while one of 50% includes one transport, either $\langle 1, 2\rangle$ or $\langle 1, 3\rangle$. The schedule we study describes a journey from Asia to Europe, (See table 6.11). We generate artificial demands by adding randomly generated lower bounds on some of the container types traveling between ports. We do this such that once the vessel is leaving the Asian trade zone, the lower bounds represent at most, 70% of the total volume capacity. The

Port	Code	D. (km)	Over. (US$)	C.P. (h)	C.R. (US$)	Zone
Qingdao	CNTAO	439	25	25	260	A
Ningbo	CNNGB	87	36	25	250	A
Shanghai	CNSHA	968	62	30	277	A
Shenzhen	CNYTN	1872	78	21	283	A
Tanjung Pelepas	MYTPP	9030	59	22	256	A
Le Havre	FRLEH	250	148	25	219	E
Rotterdam	NLRTM	256	148	20	243	E
Bremerhaven	DEBRV	109	121	25	293	E
Hamburg	DEHAM	386	224	28	277	E
Antwerp	BEANR	0	122	22	270	E

Table 6.11: *Port information of the schedule used in the multi-port setup.* First column is the port name, while the second one represents the port code. Third column is the distance in kilometers from the current port to the next one in the schedule. The fourth column is the price of a overstowing move, while the fifth column is the crane productivity, i.e., number of moves performed by a crane per hour. Seventh column is the price of a crane per hour, and the last column is the trade zone where the port belongs, i.e., A= Asia, E= Europe.

revenue for each container type is calculated based on the revenue model from the single port experiments, where the revenue of stowing a *40'* container (US$ 1,000), is replaced by the corresponding value in table 6.12. The information presented in these two tables is partially based on the benchmark introduced in Brouer et al. (2013), and interviews with container brokers. The coefficients of the remaining objectives are calculated based on the values from table 6.12. For the IP model, the coefficient for hatch and stack overstowage is read from the table, while the crane makespan coefficient is calculated based on the crane productivity, the hourly crane rate, and the number of cranes assigned to the vessel. Crane productivity and hourly crane rate have been randomly generated with values between 20 and 30 moves, and US$200 and US$300, respectively. The number of cranes assigned to a vessel in each port is six. The ballast water penalization is calculated for each port based on the distance between the current port and the next port. We read this information from table 6.11 and multiply it by a scaling constant (0.01). For the decomposition models, the probability of overstowage is 30% of the cost of the overstowage objective. For this experimental setup we consider

	CNNGB	CNSHA	CNYTN	MYTPP	FRLEH	NLRTM	DEBRV	DEHAM	BEANR
CNTAO	300	300	300	810	2990	3050	2970	3040	3020
CNNGB	-	270	280	790	2850	3000	2800	3010	3000
CNSHA	-	-	740	530	3160	2960	3070	3140	3120
CNYTN	-	-	-	410	3230	3270	3440	3180	2940
MYTPP	-	-	-	-	2820	2750	2700	2760	3350
FRLEH	-	-	-	-	-	1200	1200	1200	1200
NLRTM	-	-	-	-	-	-	390	400	1200
DEBRV	-	-	-	-	-	-	-	400	410
DEHAM	-	-	-	-	-	-	-	-	400
BEANR	-	-	-	-	-	-	-	-	-

Table 6.12: *Transportation revenue matrix for a 40' container in US$.*

schedules visiting 4 different number of ports (3, 5, 7, and 10), where the last port in the Asian trade zone, Tanjuang Pelepas, is always in the middle of the journey. We define two extra type-sets, $T^{24} = \mathcal{T}^{32} - \{(l, h, r, w) | l \in L,\ h \in H,\ r \in R,\ w \in \{6, 14\}\}$, and $T^{16} = \mathcal{T}^{24} - \{(l, h, r, 27) | l \in L,\ h \in H,\ r \in R\} \cup \{(40, HC, r, 3) | r \in R\}$, and three different transport densities: 50%, 75%, and 100%. Thus, for each of the ten container vessels from table 6.5, we generate 36 multi-port instances and solve them with the IP model and the IP and MIP decompositions. Table 6.13 summarizes the scalability of the approaches.

It is evident that the only approach that scales is the MIP decomposition, solving 91.7% of all instances. The IP decomposition managed to solve only 3% of the instances, and the IP model 0.55%. Memory seems to be a major issue for the IP model, since it affects 47.2% of the instances. This is somehow expected since the number of variables and constraints used in this model is considerably greater than that of the IP and MIP

Solver	Instances	Solved	Time out	Out of mem.	infeasible
IP	360	2	183	170	5
DMIP	360	330	29	0	1
DIP	360	13	344	3	0

Table 6.13: *Summary of the experimental results in the multi-port setup.* The first column represents the solver used to solve the instances, the second column the total number of instances, the third column the number of instances solved within 3,600s, the fourth the number of instances that timed out and, the fifth column the number of instances that ran out of memory. Finally, the last column represents the number of unfeasible instances.

decompositions and increases proportionally to the number of ports and transports. With respect to the IP decomposition, all solved instances have a schedule of 3 or 5 ports, with a tendency towards low transport density and size of type-set. All instances but one are within the last 6 vessels (smaller vessels), with 38% being instances of vessel 10. This tendency seems to indicate a possible influence of the number of variables and constraints in the problem's hardness, though it is not the only factor.

With respect to the MIP decomposition, all instances that were not solved have a schedule visiting the maximum number of ports. The 29 unsolved instances are spread over the first nine vessels, with the greatest number (7 instances), belonging to vessel 3. With respect to transport density, 82.7% of the unsolved instances have a density of 75% or more, and 55% a density of 100%. The number of types follows a similar pattern, with unsolved instances with 24 types or more representing 82.7%, and instances with 32 types representing 51.7%. Few conclusions can be drawn with respect to the IP model and IP decomposition since they did not solve many instances. Not surprising, there is tendency relating the number of variables (e.g., size of type-set, number of transports, etc.), and number of constraints (depends on the number of ports in the schedule), with the hardness of the problem for all three models. Moreover, there is a significant impact of the decomposition on the memory consumption of the solvers, since only 3 instances of the IP decomposition run out of memory.

Now we analyse the instances solved by the MIP decomposition. Figure 6.3 depicts the revenue of all solved instances, grouped by vessel and number of ports in their schedule. As expected, the tendency observed in the single-port experiments indicating the impact of the vessel size on revenue generated can also be seen in the multi-port experiments. Moreover, it is obvious that the number of ports has a direct impact on revenue. Grouping instances that are similar with respect to transport density and number of types did not show any tendencies. The left plot in Figure 6.4 depicts the impact of the number of ports in the schedule on the solution time of the instances. Though with a few outliers, we can see that for each vessel there is a tendency for the instances with the greatest number of ports to take the longest time to solve. All instances with three and five ports solved within 600s, while instances with ten ports go from 685s, up to 3,600s. Most of the instances have a gap between 0% and 0.1%,

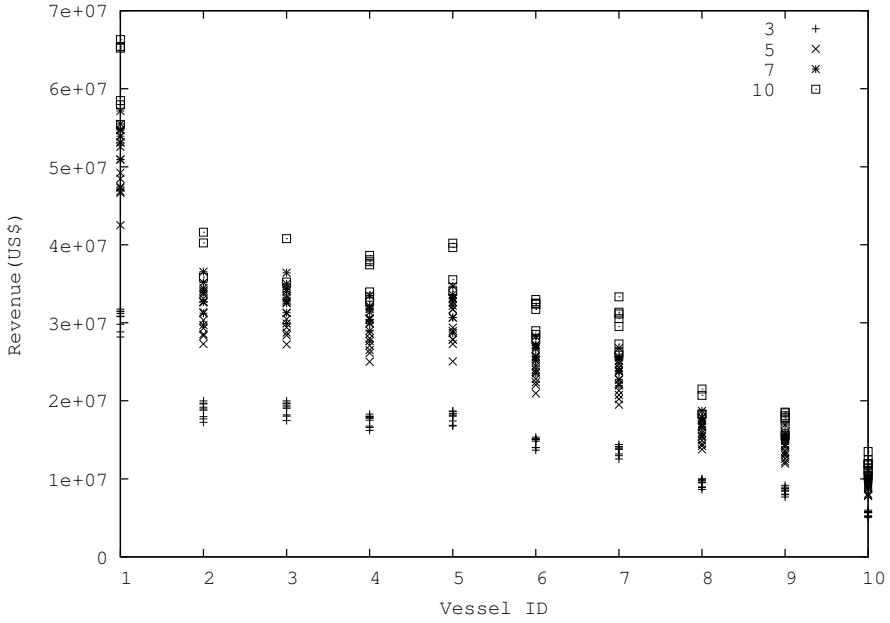


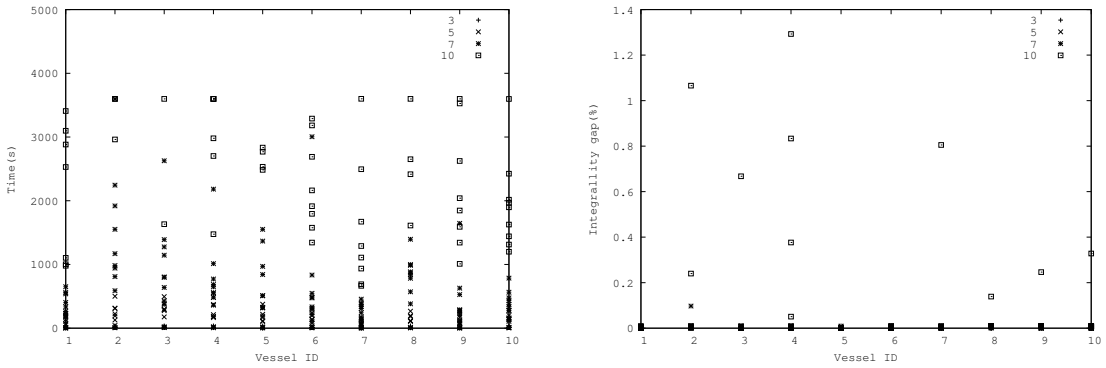
Figure 6.3: Revenue of the instances solved by the MIP decomposition, grouped by number of ports in their schedule.

a very low percentage, (See right plot in Figure 6.4). Few outliers go over 0.2%, all of them being instances with ten ports in their schedule. As with the revenue analysis, the instances were also grouped based on transport density and the size of the type-sets, but no clear tendencies were observed. In conclusion, the MIP decomposition model scales well for finding multi-port cargo compositions that optimize revenue. Moreover, there is a stronger correlation between the hardness of the problem and the number of ports in the schedule than with the vessel size (number of locations in this case), since for all the vessels most of the hard instances have 10 ports, followed by the ones with 7 ports.

6.7 Conclusions

In this chapter, we have introduced the definition of the Cargo Composition Problem, a new application of accurate stowage planning in the liner shipping domain. We have also introduced the first mathematical model with variable displacement that assigns containers to vessel slots, and that is solvable on real instances. To increase scalability, we extended the master planning model presented in chapter 4 to cope with variable displacement and use it to generate sub-optimal solutions for the CCP.

We tested our models in a benchmark suite of ten real container vessels, in a single port and multi-port scenario. In the single port setup we showed the flexibility of the models by optimizing TEU intake and intake of containers with specific features. The



(a) (b)

Figure 6.4: Instances solved by the MIP decomposition grouped by vessel and number of ports in the schedule. (a) CPU time of the instances. (b) Integrality gap.

experiments showed that for intake optimization the MIP decomposition produced very similar results to those of the IP model but only used a small fraction of computation time. When optimizing intake of containers with specific features, however, we observed small discrepancies in the solutions due to inaccuracies in the decomposition, though they can be addressed by improving the capacity constraints and increasing the accuracy of pre-computed values. Moreover, we optimized revenue and compared our results with a model from the literature, showing the impact of accurate stowage modeling. Our experiments also showed the strength of the approach, introduced to cope with variable displacement when evaluating the linear relaxations of the models. In the multi-port scenario, we have tested the impact of features such as number of ports in the schedule and number of different container types. The results showed the advantage of the decomposition with respect to the complete model in terms of memory consumption and scalability. Our decomposition successfully scaled to optimize revenue for instances up to 10 ports and solving 91.7% of all tested instances.

As future work, we plan to incorporate lashing forces to our models, a feature of stowage models that limits the cargo compositions that can be stowed in sections on deck of a vessel. Additionally, we plan to study models combining the CCP with other important problems in the liner shipping industry, (e.g., network design and vessel deployment).

Chapter 7

Conclusions

In this thesis we have studied the problem of stowing containers into container vessels and its application to different operational and commercial decisions of liner shipping companies. Specifically, we have formulated stowage optimization models with acceptable trade-offs between accuracy and scalability to assist solving stowage dependant planning problems in the liner shipping domain. Hence, the answer to the thesis question is affirmative.

For stowage planning, we presented two approaches with different trade-offs between accuracy and scalability for automatic generation of stowage plans. Both approaches are based on a 2-phase heuristic decomposition that in the first phase (master planning), assigns containers to vessel sections such that stability and stress force constraints are satisfied and time at port is minimized; while in the second phase (slot planning), containers assigned to vessel sections are stowed into slots, satisfying stacking patterns and capacities, and minimizing time at port as well as robustness of the stowage plans.

The first approach models a representative problem that stows *20'* and *40'*, reefer, and normal containers in a container vessel with release, such that stability and stress force constraints were within limits and all stacking constraints were satisfied. We presented a mathematical model for master planning that accurately represents stability and stress force constraints based on hydrostatics and includes ballast tanks, a feature of stowage planning widely disregarded in previous academic work. We focused on studying the impact of small displacement variations introduced by the ballast tanks on the accuracy on stability and stress force constraints and the center of gravity. The accuracy of the model was experimentally evaluated with industrial data and the results showed that it can successfully use ballast water to alter the stability and stress forces of a container vessel, allowing the possibility of applying stowage configurations that are otherwise infeasible.

For slot planning, we introduced an accurate representative model to stow vessel sections. We developed a CP model that, to our knowledge, is the fastest approach to generate optimal slot plans. Our CP model produced optimal solutions for most of the industrial instances where it was tested. Moreover, we successfully evaluated the CP model as a component of a module for automatic generation of stowage plans, where it

also found optimal solutions for most of the vessel sections in less than a 1s.

The second stowage planning approach was a heuristic that served as the optimization component in a commercial decision support tool used to stow container vessels interactively. For this heuristic, we reduce the model accuracy to cope with the need for fast generation of stowage plans in an interactive environment. To compensate for the reduced accuracy, our heuristic integrates the know-how of stowage coordinators, represented as user preferences, to address stability and stress force constraints and combinatorial objectives that minimize time at port. Our experiments showed that this heuristic is able to produce plans fast enough to be used in an interactive decision support tool, and that these plans are competitive with those produced by expert users.

For the cargo composition problem, we have introduced the first mathematical model with variable displacement that assigns containers to vessel slots, and is solvable on real instances. To increase scalability, we extended our accurate master planning model with the capability of dealing with variable displacement. We tested our models in a benchmark suit of real container vessels and performed analysis on intake and revenue optimization in single port and multi-port scenarios. Our experiments showed that the extended master planning model produced similar solutions to those produced by the complete model using less CPU time and hardware resources. Moreover, we optimized cargo revenue and showed the impact of accurate stowage modeling on this kind of optimization. To our knowledge, this is the first application of stowage modeling to another problem than stowage planning.

7.1 Outlook and Future Directions

With several planning problems in the liner shipping domain demanding a clear picture of the viability of stowing cargo compositions in container vessels, the necessity for developing accurate and scalable stowage optimization models is greater than ever. The stowage optimization models presented in this dissertation are a first step towards a widespread use of stowage modeling across liner shipping companies. As future work, we suggest two directions. The first relates to the accuracy and scalability of the models. We have, with our industrial collaborator, defined representative problems that consider the most relevant features of container stowage, such that the solutions produced by our models can be evaluated by expert users. To incorporate models based on these representative problems into daily operations, however, they need to be extended with additional features of the real problem, (e.g., line-of-sight and lashing forces). Thus, an interesting line of future work is to efficiently formulate the new constraints added to the representative problems and study their impact on performance and solution quality of the models presented in this thesis.

With respect to scalability, an interesting topic to investigate is the use of mathematical decompositions that can overcome the scalability problem of models assigning individual containers to slots. Though it is unlikely that these models scale for practical

use, they can be used for finding optimal solutions for large instances and allow us to evaluate the accuracy of heuristics and decompositions formulated to perform stowage optimization. Additionally, given that the models introduced in this dissertation sometimes took long time to solve, we consider anytime algorithms for stowage optimization an important research topic.

The second direction of future work is to study models combining accurate stowage modeling with other important problems in the liner shipping industry ,(e.g., network design (Brouer et al. (2013)), vessel deployment (Alvarez (2009)), fleet repositioning (Tierney (2013)), etc.). Models to tackle these problems are built with oversimplified stowage representations that assume that containers of any kind can be transported by a vessel without affecting its maximum intake, time at port, or fuel consumption. This assumption is far from reality and may reduce their applicability. Stowage models with a trade-off between accuracy and scalability, adapted to each of these problems, can improve the applicability of their solutions without significantly affecting CPU time.

Bibliography

Alphaliner (2013, 03). Cellular fleet at 1st march 2013.

Alvarez, J. F. (2009). Joint routing and deployment of a fleet of container vessels. *Maritime Economics & Logistics 11*(2), 186–208.

Ambrosino, D., D. Anghinolfi, M. Paolucci, and A. Sciomachen (2010). An experimental comparison of different heuristics for the master bay plan problem. In *Proceedings of the 9th Int. Symposium on Experimental Algorithms*, pp. 314–325.

Ambrosino, D. and A. Sciomachen (1998). A constraint satisfaction approach for master bay plans. *Maritime Engineering and Ports 36*, 175–184.

Ambrosino, D. and A. Sciomachen (2003). Impact of yard organization on the master bay planning problem. *Maritime Economics and Logistics* (5), 285–300.

Ambrosino, D., A. Sciomachen, D. Anghinolfi, and M. Paolucci (2009). A new three-step heuristic for the master bay plan problem. *Maritime Economics and Logistics 11*(1), 98–120.

Ambrosino, D., A. Sciomachen, and E. Tanfani (2004). Stowing a containership: the master bay plan problem. *Transportation Research Part A: Policy and Practice 38*(2), 81–99.

Andersen, M. W. (2010). *Service Network Design and Management in Liner Container Shipping Applications*. Ph. D. thesis.

Aslidis, A. H. (1984). Optimal container loading. Master's thesis, Massachusetts Institute of Technology.

Autoship Systems Corporation (2013). Autoload. `http://cargomanagement.autoship.com`.

Avriel, M., M. Penn, and N. Shpirer (2000). Container ship stowage problem: complexity and connection to the coloring of circle graphs. *Discrete Applied Mathematics 103*, 271–279.

Avriel, M., M. Penn, N. Shpirer, and S. Witteboon (1998). Stowage planning for container ships to reduce the number of shifts. *Annals of Operations Research 76*, 55–71.

Barrass, B. and C. D. Derrett (2011). *Ship stability for masters and mates*. Butterworth-Heinemann.

Botter, R. and M. Brinati (1992). Stowage container planning: A model for getting an optimal solution. In *Proceedings of the 7th Int. Conference on Computer Applications in the Automation of Shipyard Operation and Ship Design*, pp. 217–229.

Brouer, B. D., J. F. Alvarez, C. E. Plum, D. Pisinger, and M. M. Sigurd (2013). A base integer programming model and benchmark suite for linear shipping network design. *Transportation Science*.

Christiansen, M., K. Fagerholt, B. Nygreen, and D. Ronen (2007). Maritime transportation. *Handbooks in Operations Research and Management Science 14*, 189–284.

Davidor, Y. and M. Avihail (1996). A method for determining a vessel stowage plan, Patent Publication WO9735266.

Delgado, A. and R. M. Jensen (2013). Cargo composition analysis in container vessels. To be submitted to European Journal of Operational Research.

Delgado, A., R. M. Jensen, and N. Guilbert (2012). A placement heuristic for a commercial decision support system for container vessel stowage. In *Informatica (CLEI), 2012 XXXVIII Conferencia Latinoamericana En*, pp. 1–9. IEEE.

Delgado, A., R. M. Jensen, K. Janstrup, T. H. Rose, and K. H. Andersen (2010, October). A constraint programming model for fast optimal stowage of container vessel bays. Technical Report TR-2010-133, IT University of Copenhagen.

Delgado, A., R. M. Jensen, K. Janstrup, T. H. Rose, and K. H. Andersen (2012). A constraint programming model for fast optimal stowage of container vessel bays. *European Journal of Operational Research 220*(1), 251 – 261.

Delgado, A., R. M. Jensen, and C. Schulte (2009). Generating optimal stowage plans for container vessel bays. In *Proceedings of the 15th Int. Conference on Principles and Practice of Constraint Programming (CP-09)*, Volume 5732 of *LNCS Series*, pp. 6–20.

Delgado, A., D. Pacino, and R. Jensen (2013). Container vessel stowage planning with ballast tanks. In *Proceedings of the 2nd INFORMS Transportation Science and Logistics Society Workshop*.

Drewry Maritime Research (2011). Container census 2011: Survey and forecast of global container units.

Dubrovsky, O., G. Levitin, and M. Penn (2002). A genetic algorithm with a compact solution encoding for the container ship stowage problem. *Journal of Heuristics 8*, 585–599.

Feng, C.-M. and C.-H. Chang (2008). Optimal slot allocation in intra-asia service for liner shipping companies. *Maritime Economics & Logistics 10*(3), 295–309.

Flor, M. (1998). Heuristic algorithms for solving the container ship stowage problem. Master's thesis, Technion, Haifa, Isreal.

Gecode Team (2006). Gecode: Generic constraint development environment. Available from `http://www.gecode.org`.

Giemsch, P. and A. Jellinghaus (2004). Optimization models for the container ship stowage problem. In *Operations research proceedings 2003: selected papers of the International Conference on Operations Research (OR 2003), Heidleberg, September 3-5, 2003*, pp. 347. Springer Verlag.

Guilbert, N. and B. Paquin (2010). Container vessel stowage planning, Patent Publication US2010/0145501.

Gumus, M., P. Kaminsky, E. Tiemroth, and M. Ayik (2008). A multi-stage decomposition heuristic for the container stowage problem. In *Proceedings of the Manufacturing and Service Operations Management (MSOM) conference*.

Hamburg, S. V. (2008). *Storck Guide: Stowage & Segregation to IMDG Code Including Amendment 34-08*. Storck guide. Storck.

INTERSCHALT maritime systems AG (2013). Stowman. `http://www.interschalt.de/en/is-seacos-stowage-planning.htm`.

Kang, J. and Y. Kim (2002). Stowage planning in maritime container transportation. *Journal of the Operations Research Society 53*(4), 415–426.

Kjeldsen, K. H. (2012). *Liner shipping network design, routing and scheduling*. Ph. D. thesis.

Li, F., C. Tian, R. Cao, and W. Ding (2008). An integer linear programming for container stowage problem. In M. Bubak, G. van Albada, J. Dongarra, and P. Sloot (Eds.), *Computational Science ICCS 2008*, Volume 5101 of *Lecture Notes in Computer Science*, pp. 853–862. Springer Berlin / Heidelberg.

Muller+Blanck Software GmbH (2013). Capstan3. `http://www.capstan3.com/capstan3.html`.

Navis (2013). Navis PowerStow. `http://www.navis.com/solutions/shipping`.

Nugroho, S. (2004). Case-based stowage planning for container ships. In *International Logistics Congress*, Number December 2004, pp. 2–3.

Pacino, D. (June 2012). *Fast Generation of Container Vessel Stowage Plans - Using mixed integer programming for optimal master planning and constraint based local search for slot planning*. Ph. D. thesis, IT University of Copenhagen.

Pacino, D., A. Delgado, R. Jensen, and T. Bebbington (2012). An accurate model for seaworthy stowage planning with ballast tanks. pp. 17–32. Springer. Proceedings of the Third International Conference on Computational Logistics, (ICCL12).

Pacino, D., A. Delgado, R. M. Jensen, and T. Bebbington (2011). Fast generation of near-optimal plans for eco-efficient stowage of large container vessels. In J. Bse, H. Hu, C. Jahn, X. Shi, R. Stahlbock, and S. Vo (Eds.), *Computational Logistics*, Volume 6971 of *Lecture Notes in Computer Science*, pp. 286–301. Springer Berlin / Heidelberg.

Pacino, D. and R. M. Jensen (2012). Constraint-based local search for container stowage slot planning. In *Proceedings of the International Multi-Conference of Engineers and Computer Scientists*, pp. 1467–1472.

Pesant, G. (2004). A regular language membership constraint for finite sequences of variables. In *Proceeding of Principles and practice of constraint programming*, Volume 3258 of *Lecture Notes in Computer Science*, pp. 482–495. Springer.

Sciomachen, A. and A. Tanfani (2003). The master bay plan problem: a solution method based on its connection to the three-dimensional bin packing problem. *IMA Journal of Management Mathematics 14*, 251–269.

Scott, D. and D. Chen (1978). A loading model for a container ship. Technical report, Matson Navigation Company, Los Angeles.

Shaw, P. (2004). A constraint for bin packing. In *Proceeding of Principles and practice of constraint programming*, Volume 3258 of *Lecture Notes in Computer Science*, pp. 648–662. Springer.

Smith, B. (2006). Modelling. In F. Rossi, P. van Beek, and T. Walsh (Eds.), *Handbook of Constraint Programming*, Chapter 11. Elsevier.

Stopford, M. (2009). *Maritime Economics* (Third ed.). Routledge.

Tierney, K. B. (2013). *Optimizing Liner Shipping Fleet Repositioning Plans*. Ph. D. thesis, IT University of Copenhagen.

Ting, S.-C. and G.-H. Tzeng (2004). An optimal containership slot allocation for liner shipping revenue management. *Maritime Policy & Management 31*(3), 199–211.

UNCTAD (2012). Review of maritime transport.

Van Hentenryck, P. and J. P. Carrillon (1988). Generality vs. specificity: an experience with AI and OR techniques. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 660–664. ACM press.

Webster, W. C. and P. Van Dyke (1970). Container loading. A container allocation model: I - introduction background, II - strategy, conclusions. In *Proceedings of Computer-Aided Ship Design Engineering Summer Conference*. University of Michigan.

Wilson, I. and P. Roach (1999). Principles of combinatorial optimization applied to container-ship stowage planning. *Journal of Heuristics 5*, 403–418.

Wilson, I. D. and P. Roach (2000). Container stowage planning: A methodology for generating computerised solutions. *Journal of the Operational Research Society 51*(11), 248–255.

Yoke, M., H. Low, X. Xiao, F. Liu, S. Y. Huang, W. J. Hsu, and Z. Li (2009). An automated stowage planning system for large containerships. In *Proceedings of the 4th Virtual Int. Conference on Intelligent Production Machines and Systems*.

Zhang, W., Y. Lin, and Z. Ji (2005). Model and algorithm for container ship stowage planning based on bin-packing problem. *Journal of Marine Science and Application 4*(3).

Zurheide, S. and K. Fischer (2012). A revenue management slot allocation model for liner shipping networks. *Maritime Economics & Logistics 14*(3), 334–361.