

Optimization Algorithms for Maritime Terminal and Fleet Management

A Thesis
Presented to
The Academic Faculty

by

José Fernando ÁLVAREZ

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Graduate Program in Economics, Management and Finance
Universitat Pompeu Fabra
June 2008

Optimization Algorithms for Maritime Terminal and Fleet Management

Approved by:

Daniel Serra, Committee Chair

Member of the Doctoral Committee

Helena Ramalhinho, Advisor

Member of the Doctoral Committee

Member of the Doctoral Committee

Member of the Doctoral Committee

Date Approved: _____

Dipòsit legal: B.54962-2008
ISBN: 978-84-692-0975-2

To Stephanie

Acknowledgements

I am grateful to my advisers – Professor Helena Ramalhinho, Professor Stefan Voß, and Professor Daniel Serra – for their support throughout the three years of my doctoral studies. Professor Ramalhinho allowed me to work autonomously from the start. This freedom was a source of considerable satisfaction, and has resulted in my ability to initiate and carry through with original research.

My work in maritime logistics stems from a very deliberate drive to commit to a specific field of knowledge. Professor Pankaj Ghemawat has said that “strategy means commitment”, and this has been my mantra over the last three years. I can’t overemphasize the benefits that I have derived from this observation. I am thankful to Professor Enric Ricart for introducing me to the field of corporate strategy. I am also thankful for Professor Peter Lorange’s work, which informed my decision on *how* to commit.

Of course, I was fortunate enough to have several years of professional experience before I started my doctoral studies, which provided me with a clear picture of what I wanted to do. My previous work experience left me enamored of optimization. This experience also gave me some insight into what problems were relevant, and how to approach them. In this sense, I am thankful for the influence of Dr. Spyridon Kontogiorgis, Professor Russell Rushmeier, Mark Reynolds, and Professor Linda Nozick.

When I entered the graduate program, I had been exposed to several applications in telecommunications, marketing, air transportation, and ground logistics. Unfortunately, these applications were either completely oversubscribed by other academics, slightly discredited, ignored in business circles, or not entirely interesting to me. When I moved to Barcelona, the proximity to the Mediterranean inspired my interest in all things maritime. I quickly observed the dearth of academic research on maritime optimization, and a clear picture of my research goal appeared.

Much academic work is supported by an army of unsung heroes: the taxpayers. I am

forever grateful for the financial assistance I received from Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR), a publicly funded agency of the Government of Catalonia. The fellowship I obtained from AGAUR reduced my teaching duties by half, enabling me to concentrate on my research and graduate somewhat quickly.

Throughout these years, I had the privilege of working with many talented individuals in the maritime sector. I am thankful to Carlos Lara, Capt. Josep Ollés, and the staff at TERCAT. I also enjoyed working with Chris Casey, Dr. Mehmet Ayik, Dr. Hakan Golbasi, Robert Inchausti, and all the staff at Navis LLC. My discussions with Dr. Alessandro Carrotta, Dr. Claude LePape, Professor David Simchi-Levi, and others at ILOG S.A. have helped me understand the market for optimization software.

The research included in this dissertation has been presented at three international conferences: MT Barcelona 2006, INFORMS Seattle 2007, and IAME Dalian 2008. I am grateful for all the positive feedback, suggestions for improvement, and encouragement I received from the attendees. In particular, I benefited from detailed feedback provided by Professor Luis Gouveia, Professor Michael Bell, and Professor Rommert Dekker. Professor Ellis Johnson also provided some insights during iMATH 2008. Professor Kjetil Fagerholt provided valuable feedback on my fleet routing paper. Naturally, I assume all responsibility for any remaining errors or omissions.

Finally, I thank Stephanie, Fred, and Jane for all their love and patience. Their support has been vital.

Table of Contents

Acknowledgements	iv
List of Tables	viii
List of Figures	ix
1 Introduction	1
2 Joint Routing and Deployment of a Fleet of Container Vessels	7
2.1 Literature review	9
2.2 Mathematical model	11
2.2.1 MIP formulation	14
2.3 Solution technique	18
2.4 Case study	23
2.5 Conclusions	27
3 A Lagrangean Relaxation Approach to the Vessel Planning Problem 29	
3.1 The load planning process	30
3.2 Literature review	34
3.3 Mathematical model	36
3.3.1 MIP formulation	37
3.3.2 Lagrangean relaxation	40
3.4 Solution methodology	42
3.4.1 The branch and bound framework	42
3.4.2 The subgradient optimization algorithm	45
3.4.3 Network algorithms	45
3.5 Computational results	48
3.6 Conclusions	52
4 A Heuristic Approach to the Vessel Planning Problem	53
4.1 A heuristic for efficient vessel planning	54
4.2 A decision support system for vessel planning	60
4.3 Experimental results and analysis	63
4.4 Conclusions and directions for further research	65

5	Conclusions and Future Work	68
5.1	Opportunities through 2018	68
5.1.1	Better integration of maritime transport in the supply chain	68
5.1.2	Revenue management	69
5.1.3	Focus on the environment	71
5.1.4	New patterns in world trade	72
5.1.5	Focus on safety	73
5.1.6	Arctic exploration and the Northwest Passage	73
5.1.7	Speculation in the shipping sector	74
5.2	The challenges	75
5.2.1	The complexity of the models impedes their adoption	75
5.2.2	User interfaces: less is more	75
5.2.3	The duration and cost of O.R. projects	76
5.2.4	Uncertainty in the maritime sector	77
5.2.5	Underlying mathematical complexity	77
	References	79

List of Tables

Table 1	Available vessel types	24
Table 2	Performance benchmark using ten containers	49
Table 3	Performance benchmark using fifteen containers	50
Table 4	Estimation of the optimality gap for problems with 172 containers	51

List of Figures

Figure 1	Schematic representation of the two-tier solution approach.	20
Figure 2	Crude oil prices 2007- 2008 (The Wall Street Journal, 2008)	23
Figure 3	Transportation demand at each port	25
Figure 4	Sample branch and bound tree for a single reach stacker filling three slots	43
Figure 5	Shortest path problem in variables A_{rpg}	47
Figure 6	Sequence of generic containers for two crews	56
Figure 7	Modifying a plan to generate neighbors	58
Figure 8	Initial screen of the DSS	61
Figure 9	Full-ship solution screen of the DSS	62
Figure 10	Objective function trajectory for 4,500 TEU ship	64

Chapter 1

Introduction

Over the last three decades, international trade has been a major driver of economic growth and development (UNESCAP, 2007). The rise of globalization – whereby geographically distant economic activities become closely intertwined – has been an essential driver of the increase in living standards that the world has witnessed during recent years (Wolf, 2004). Today, complex supply chains connect raw materials and manufacturers to rapidly expanding and increasingly sophisticated consumer markets.

There are many critics of globalization, and we would not presume to dismiss their claims. On the contrary, we recognize the gravity of the social and environmental challenges posed by globalization. For instance, rampant consumerism, which has been an important component of economic growth in the West, is rapidly spreading to Eastern countries. The voracious appetite – of developed and developing economies alike – for raw materials has placed enormous pressure on the natural environment, and has resulted in the overexploitation of forests, fisheries, and arable land. The increased utilization of fossil fuels and livestock has been a major driver of global warming, which will in turn have a disproportionate effect on the poor. Dismal employment conditions in many of the countries that supply the cheap goods we consume in Europe and North America cast a dark shadow over the manufacturers in question. Furthermore, the ethical values of the large corporations, which appear to derive great benefits from globalization, are often questionable.

While remaining keenly cognizant of the arguments on both sides of the debate on globalization, it is not our place to address such concerns here. Rather, we believe that globalization is a powerful force that will move forward – perhaps inevitably – over the next several decades. Whether this force is used for the general benefit of humankind and the environment around us, or rather, to the detriment of both, is entirely up to us. To a large extent, it will be the responsibility of leaders in the spheres of politics, business, and academia to determine the policies that will lead us to sustainable and fair growth.

Maritime transport is the backbone of international trade and globalization. Today, over 80% of global trade (by volume) is transported by ship (DNV Research and Innovation, 2007). Consequently, the shipping industry is closely linked to the challenges and opportunities posed by globalization – maritime transport is both a cause and a consequence of globalization.

On one hand, inexpensive, efficient, and reliable maritime transportation is a prerequisite to global trade at the scale we witness today. Indeed, spatial and economic proximity to large shipping ports has become a definitive competitive factor for manufacturers. Ocean-based transportation has become so inexpensive that it is now economically practical to import lemons from Argentina into Spain, where locally grown lemons rot on the ground (Rosenthal, 2008). It is less expensive to ship a bottle of wine from France to New York than it is to transport it overland from California. It is commonplace to ship semi-manufactured goods several times, throughout the manufacturing process, between plants that exploit tremendous economies of scale and specific technological capabilities. For a large category of products, the benefits derived from lower wages and economies of scale far outweigh the associated transportation and inventory holding costs.

On the other hand, maritime transportation as we know it today depends to a large extent on the presence of open markets and international cooperation treaties. Shipping is a truly global business. Not only are the main tangible assets of shipping companies generally in international waters, but suppliers and workers from a large number of nationalities are involved in a ship's daily operation. For instance, the ship's top officers may be from a Northern European country, while the crew are Filipino or Indian. The ship is likely

to be managed in Greece, insured in England, flagged out to Panama or Liberia, and trade between Japan and the West Coast of the United States. The ship hull will have been manufactured in South Korea or Japan, based on engineering drawings produced in Scandinavia, and fitted with a German engine.

In recent years, we have also seen the rise of global stevedoring conglomerates, such as Hutchison Port Holdings, PSA, A.P. Moller Terminals, or Dubai Ports World (DPW), which operate terminals in dozens of countries. The ability of these companies to operate internationally is critically dependent upon a degree of cooperation and openness, as was illustrated in 2006 when the Congress of the United States voted to block DPW – which is based in the United Arab Emirates – from operating maritime terminals in the United States.

There is a widespread sentiment among academics and practitioners that the maritime industry lags behind other business sectors in its ability to develop and apply advanced quantitative methods to its management practices. In particular, as will be shown in Section 2.1, it appears that the maritime transportation sector has fallen behind other modes of transportation in this respect. Christiansen, Fagerholt, Nygreen, and Ronen (2007) list several factors that have contributed to the scarcity of research into maritime planning problems: low visibility, lack of a standardized structure to the problems in the sector, an unusually high degree of uncertainty, deeply rooted traditions, and fragmentation of the sector into many small players.

We believe that there is another element which has contributed significantly to the lack of advanced and innovative managerial practices in the maritime sector. It appears that many companies have a policy of promoting former seafarers to top managerial posts. This observation is informally supported by an examination of the requirements posted by the main recruiting agencies and job boards in this sector (Spinnaker Consulting, 2008; Lloyd's List, 2008; Faststream UK, 2008). We believe that this practice limits the flow of fresh ideas into the industry, discourages top professional managers from considering a career in the sector, and is unlikely to result in merit-based and transparent hiring and promotion processes.

To advance in the global market, maritime companies will need to develop core competencies in supply chain management, logistics and information technology. We are not about to contest the importance of leveraging the knowledge of seasoned seafarers and terminal planners. On the other hand, it is not clear that a career as a vessel officer prepares the individual to implement the most effective managerial principles onshore. In the future, an important competitive advantage will be obtained by companies that can attract top managerial talent with a privileged understanding of the aforementioned fields. A key issue will be the establishment of productive relationships between these managers and experienced seafarers.

The shipping sector is now at a critical junction. The impact of the sector on the environment will fall under closer scrutiny as additional emphasis is placed on reducing vessel emissions and spills, as well as improving safety and labor standards. On the other hand, sophisticated shipping customers, who have embraced advanced supply chain management strategies themselves, will increasingly look for transport companies with correspondingly advanced capabilities. While demand for ocean-based transport is forecast to continue increasing for decades, and vessel supply is expected to remain generally tight (DNV Research and Innovation, 2007), the shipping sector will remain fiercely competitive. We believe that sophisticated mathematical modeling tools, such as those presented in this research, will confer a strong competitive advantage to companies that embrace them.

The three papers contained in this dissertation address important questions faced by liner and stevedoring companies. We have intentionally covered both types of companies in an effort to provide a broader view of the maritime sector. Similarly, we have addressed tactical as well as operational problems.

The research presented in Chapter 2 addresses a key question faced by liner companies. Given a forecast of transportation demand over a tactical planning horizon, how should the company design its services and allocate vessels to these services, in such a way that its costs are minimized, while fulfilling the expected demand? Clearly, this is an extremely important question. A large portion of the liners' operational expenditures arise from fuel consumption and terminal fees. These costs are in turn determined by the choice of

routes and transshipment points and volumes. The question is also extremely complex due to the vast number of possible routes, and the interaction between vessel routing, fleet deployment, vessel speeds, and transshipment volumes. We believe that any attempt to resolve this problem manually would result in a significantly suboptimal operation and unnecessarily high costs. We therefore propose a mixed integer programming model of the problem and an algorithm to resolve it. The proposed algorithm decomposes the problem into two tiers. The top tier generates new routes and deploys vessels using a heuristic search mechanism and based on reduced costs obtained from the lower tier. The lower tier, which is solved exactly, determines the container flows along the different routes, as well as ideal transshipment points and volumes.

In Chapter 3, we present a mathematical model of the vessel planning problem. The model considers disutility from the travel of container handling equipment (CHE) in the yard; rehandling; and vessel instability. The presence of a large number of binary variables makes this problem intractable even for instances that are two orders of magnitude smaller than those encountered in practice. We propose a Lagrangean relaxation of the problem, which results in three subproblems: a shortest path problem, an assignment problem, and a trivial minimization problem. The resulting algorithm solves problems of relevant dimensions very quickly. This speed can be attributed to two factors. The first factor is the efficiency of the network algorithms used to solve the three subproblems. The second factor is the generation of a large number of high quality primal feasible solutions that are easily derived from the subgradient optimization algorithm used to coordinate the three subproblems.

In Chapter 4, we address the vessel planning problem using tabu search. This research predates a more sophisticated heuristic algorithm that was developed for a terminal operating software vendor, and which is protected by an intellectual property agreement. The algorithm presented in this dissertation is found to produce adequate solutions very quickly.

Each of the aforementioned chapters include sections on conclusions and directions for further research. Rather than reiterating these points, Chapter 5 provides an outlook for how optimization may evolve within the maritime sector over the next decade. Our focus

is on ascertaining the type of problems that will be relevant, as well as the challenges that we will face when implementing new models and algorithms in a business environment.

Chapter 2

Joint Routing and Deployment of a Fleet of Container Vessels

Large liner companies operate scores of vessels and provide regular transportation between distant markets using a complex array of services. The cost of operating the vessel fleet is a central concern of the liner companies, and the liner's ability to control these costs constitutes a major competitive advantage. At the tactical level, the key decisions that the company makes in this sense are the determination of the routes that will be served and the number and types of vessels that will be assigned to each route. The complexity of these decisions cannot be overstated. A comprehensive solution to the problem should consider, among other factors: the demand for and pricing of transportation between different ports and geographic regions; the different operational characteristics (speed, fuel consumption, crew requirements, and carrying capacity) of each vessel in the fleet; level of service requirements; conference agreements; and transshipment ports and costs.

A vast body of scientific literature has been devoted to fleet sizing, deployment, and routing problems in air (e.g. Gopalan and Talluri, 1998; Rushmeier and Kontogiorgis, 1997), truck (e.g. Simchi-Levi, Chen, and Braca, 2005), and rail (e.g. Sherali and Maguire, 2000) transportation. By comparison, little research is available regarding the methods employed by maritime vessel operators in making the corresponding decisions. The recent work by Fagerholt and Lindstad (2007) in tramp and industrial shipping constitutes a welcome

advance in the application of sophisticated algorithms to the vessel scheduling problem. In the case of liner shipping, however, little progress has been made in recent years.

The fact that the liner shipping sector has not more thoroughly embraced the use of advanced mathematical programming algorithms is perplexing, given the success reported in other complex and capital intensive transportation sectors. For instance, Sherali and Maguire (2000) report benefits of over half a billion dollars to railroad companies using a fleet sizing algorithm. Armacost, Barnhart, Ware, and Wilson (2004) describe a network planning tool that generated savings of more than \$87 million at UPS over a two year period. Mason, McKenney, Carlson, and Copeland (1997) describe the essential role of mathematical programming models at FedEx. Rushmeier and Kontogiorgis (1997) reported annual savings of \$15 million from the use of their models at USAir. Abara (1989) describes the use of mathematical programming models for fleet assignment at American Airlines. Closer to the liner sector, Fagerholt and Lindstad (2000) reported annual savings of US\$7 million following the implementation of a vessel scheduling algorithm for a tramp shipping operation along the Norwegian Coast. Today, it would be unthinkable to operate a large airline or ground logistics company in the absence of advanced mathematical programming tools for fleet sizing and scheduling.

In this paper we present a mixed integer programming (MIP) formulation for the joint optimization of vessel routing and deployment. As will be seen in Section 2.1, most previous research has treated the routing and deployment problems separately. In most cases, the fleet sizing models assume that a set of good candidate routes has been obtained, either from past experience or through a separate process. One of the central goals of this research is to demonstrate a model that can consider the routing and deployment aspects concomitantly. The proposed model permits the representation of multiple vessel types, each with a particular profile of fuel consumption, daily running costs, and carrying capacity. The model can represent the transshipment of containers, either at specific ports selected by the user, or at ports suggested by the model. The determination of target vessel speeds for each service is also considered within the model.

The remainder of this paper is organized as follows. The next section provides a brief

overview of related scientific literature. Section 2.2 presents a mathematical model of the vessel routing and deployment problem that captures the principal tactical-level decisions and costs faced by a global container carrier. Section 2.3 presents an algorithm to solve large instances of the proposed model. Section 2.4 provides a case study, where we apply the model to a putative liner company, and examine the effects of changes in bunker prices to the routing strategy of this company. The final section presents conclusions and directions for further research.

2.1 Literature review

Notteboom (2006) provides a descriptive discussion of the parameters considered by liner companies when designing a service schedule, as well as the methods used to correct delays when they do occur.

Christiansen et al. (2004, 2007) provide a review of recent research in ship routing and scheduling. They classify the literature according to the type of shipping operation (industrial, tramp, or liner shipping) and the planning level addressed (strategic, tactical, or operational). The authors point out several key differences between five modes of freight transportation, namely: ship, air, truck, pipeline, and rail. An important observation made by the authors is that research of routing and scheduling methods in the maritime sector has lagged far behind corresponding research in the air- and land-based modes. In particular, they note the paucity of research in liner vessel routing, an observation that is also made by Sherali and Al-Yakoob (2006)

Rana and Vickson (1991) present an arc-flow formulation of the vessel routing and deployment problem. Inasmuch as the model simultaneously addresses the routing and deployment problems for a fleet of heterogeneous liner vessels, this model is possibly the closest precursor to the one we propose. A large number of integer variables is required for a general arc-flow formulation, which generally poses a significant computational challenge. The authors mitigate this complexity by imposing restrictions on the type of routes that can be constructed by the model. The resulting network structure is not representative of liner business practice today. Furthermore, the authors use several nonlinear constraints

in their formulation, which constitute an additional difficulty. The authors resolve these difficulties using decomposition techniques, and arrive at near-optimal solutions.

Cho and Perakis (1996) propose a linear programming formulation for the vessel deployment problem. The model assumes that the number of vessels available to the company is fixed over the planning horizon. An explicit enumeration of the routes that may be employed is required – the model cannot generate new routes. A separate MIP model is proposed to address the liner company’s capital investment decisions over longer time periods.

Powell and Perakis (1997) develop an integer programming model for optimal liner fleet deployment. The model captures operating costs for each vessel-route combination, lay-up costs, vessel-route incompatibilities, and required service frequencies. An enumeration of all potential routes is required. The model determines the number and type of vessels to serve each route, as well as the number of vessels that are laid up.

Fagerholt (1999) presents a fleet sizing algorithm for a shipping operation along the Norwegian Coast. The transportation operation is naturally divided into a feeder portion and a distribution portion, which are in turn connected through a single depot. The problem is solved by generating a list of feasible routes, and then assigning vessels to routes using a set partitioning formulation.

Fagerholt and Lindstad (2000) address the problem of finding an efficient fleet composition and schedule for delivering supplies from a single onshore depot to multiple offshore installations in the North Sea. Cargoes of different types are segregated into different compartments and the deck. Limited deck space creates a constraint on the number of facilities that can be served during a single sailing of a vessel. There are additional constraints on the times at which a vessel can unload its cargo at the different offshore facilities. The proposed algorithm operates in two phases. The first phase generates a list of feasible schedules for each vessel. Given the small dimension of the instances addressed, it is possible to generate all feasible schedules for every vessel in the fleet. The second step of the solution procedure uses the aforementioned schedules to generate the constraint matrix of a mixed integer program. This program obtains the fleet composition and schedules that minimize operating costs and total sailing time.

Song, Zhang, Carter, Field, Marshall, Polak, Schumacher, Sinha-Ray, and Woods (2005) examine the cost-efficiency of the global container shipping network. Their heuristic algorithm assigns trade volumes to the actual set of liner services available worldwide as of the year 2002. The vessel routing and deployment decisions are part of the input data, so the methodology focuses on solving the multicommodity flow problem with transshipment costs.

Fagerholt and Lindstad (2007) present a decision support system (DSS) for ship routing and scheduling in industrial and tramp shipping. The DSS allows for detailed consideration of business requirements, including representation of different vessel types; compatibility between products and vessels; port and canal fees and operating hours; multiple possible objective functions; and spot charter rates. The DSS can be used as a manual planning tool to evaluate different scenarios, or it can generate schedules automatically using a multi-start local search heuristic. This heuristic generates a large number of initial solutions, and then improves them using intra-route and inter-route cargo-vessel reassignments. The proposed DSS has been deployed to several shipping companies, which have in turn derived significant benefits from its use. Unfortunately, the proposed formulation is specifically adapted to industrial or tramp shipping, where the business model differs substantially from the one considered in the present research.

2.2 Mathematical model

The purpose of the model is to minimize the operating expenses of a liner company over a tactical planning horizon. Our model does not consider the strategic question of vessel purchase and retirement, as the planning horizon for these decisions is typically between twenty and thirty years. Furthermore, vessel acquisitions and sales are governed by extremely volatile market forces (Stopford, 1997), which have the potential to generate large distortions inside a tactical-level model. Vessel demolition should be considered a strategic-level decision, and this alternative is similarly excluded from the model. Therefore, we assume that the liner company already owns a fleet of vessels, and is bound to a schedule of payments on the principal price of the vessels. These payments must be made regardless of whether the vessels are in service or laid up, and are therefore not included in our model.

Another plausible alternative, that of chartering out some vessels temporarily, is readily represented in the proposed model.

Rather than represent the operational characteristics of every possible vessel, we group similar vessels using a small number of generic vessel types. The relevant characteristics of each vessel type include carrying capacity in TEU¹; fuel consumption as a function of speed; and daily running costs (DRC) – crew, repairs and maintenance, oil and lubricants, protection and indemnity, and stores (Alderton, 2005). Each vessel type has a small set of possible operating speeds. The model selects a speed for each vessel, and it is assumed that the vessel maintains this speed throughout all segments of its service. It should be noted that cruising speed selection is generally considered an operational-level problem. We propose an evaluation of speed selection at the tactical level, since this parameter has a direct impact on the number of TEU-miles each service can provide. Naturally, speed corrections will always occur at the operational level, based on immediate circumstances.

We assume that the liner company has manifest demand for transportation between ports worldwide. This demand appears at a constant rate throughout the planning period. In practice, liner companies will be subject to seasonal variations in demand. We therefore assume that the planning horizon considered in the model is restricted to a period where demand for transportation is indeed fairly stable.

Rather than considering the true origin and destination of the merchandise in the hinterland, we aggregate all demand for transportation at the maritime port serving the associated hinterland. We assume that the liner company can choose not to transport any portion of the demand, but incurs lost revenue and perhaps an additional monetary penalty (as a proxy to loss of goodwill) for each unit of manifest demand that it fails to transport. In order to support some heuristic search mechanisms, we group the ports into a small number of regions (e.g. Western Mediterranean, North America West Coast, etc.). The cost of loading and unloading a TEU at every port in the model is known.

¹A TEU, or twenty-foot equivalent unit, is a standard container: 20 ft long, 8 ft wide, and 8.5 ft tall. The TEU is the universally accepted unit for measuring the capacity of container ships and the activity of container terminals. The 40-ft container, which is also widely used, is counted as two TEUs, and is also referred to as a FEU.

The travel time between any two ports is assumed to be deterministic. The amount of time that each type of vessel spends at each port can be estimated with reasonable accuracy and is independent of the number of containers being lifted during the call. This assumption is satisfactory in the presence of a significant fixed time component to each port call (due to the vessel approach, mooring, etc.) and when vessels of similar capacity lift a similar number of containers each time they call at a particular port.

The model can accommodate routes that are proposed externally by a planner, as well as routes that are generated by the solution algorithm. We impose no restrictions on the routes that are added by the planner, and only minimal restrictions on the types of routes that the algorithm can generate. In the latter case, every route generated by the algorithm must be a loop, in the sense that every vessel assigned to that route will visit the same ordered sequence of ports during each voyage. We also limit the duration of each voyage in order to comply with crewing regulations or agreements. Beyond these two limitations, we allow the model to utilize any route that would result in a reduction of operational costs. In contrast to the work of Bendall and Stent (2001), a hub-and-spoke network will be employed only if and where it results in reduced overall costs.

By default, it is assumed that an unlimited number of containers can be transhipped at any port. The model will select the transshipment locations that result in lower costs overall. When a liner company has existing agreements at specific transshipment hubs, the lower lift costs and port times at those locations should naturally induce the model to tranship at these locations. If needed, the transshipment volumes at any port can be controlled by imposing upper bounds on the appropriate flow variables.

Draft, length, and width restrictions at ports, canals, or route segments can be represented in the model by disabling any vessel type-arc combination as needed. Land bridges and transportation to land-locked countries can be represented using the entities in the proposed model.

Some of the modeling assumptions presented above are appropriate at the tactical level, but are unrealistic within an operational timeframe. It is understood that the liner company must also develop operational-level policies to address temporary variations in its business

environment.

2.2.1 MIP formulation

In this section, the tactical fleet sizing and routing problem is formulated as a MIP. In the maritime sector, a service is defined as a sequence of ports to be visited following a published schedule. While we adhere to this convention, the core entity of our model must be more specific in order to capture all transshipment costs properly. We therefore define a *run* as a particular vessel type-speed-service combination. The following sets are used in the formulation:

Π	All ports in the model.
Λ	All regions in the model.
Ξ	All runs in the model.
$\Xi_{i,j}$	Set of all runs incident upon arc (i, j) .
Δ	Set of all possible arcs in the model, $\Pi \times \Pi$. All arcs are directed and uncapacitated.
Δ_r	Set of arcs used in run r .
Υ	Vessel types in the model.

The following parameters represent the known data for the problem:

v_r	Vessel type used in run r .
s_r	Speed of all vessels used in run r .
c^v	Capacity (in TEU) of vessels of type v .
f^v	Daily running costs (DRC) for vessels of type v over the entire planning horizon.
\tilde{f}^v	Cost or revenue of excluding a vessel of type v from operations. This can be the revenue obtained from chartering the vessel out over the entire planning horizon, or the cost of laying up the vessel, depending on transportation market conditions.

g^{vs}	Fuel consumption (tons per mile) for vessels of type v steaming at speed s .
h^v	Fuel consumption (tons per day) for vessels of type v when idle at port.
e	Fuel price per ton.
z^v	Number of vessels of type v available.
l_{ij}^v	Length in nautical miles of a direct sailing from port i to port j using vessel type v .
b_r	Length (nautical miles) of run r ; $b_r = \sum_{\{(i,j) \in \Delta_r\}} l_{ij}^{v_r}$.
m_r	Number of round trips performed by a vessel on run r during the planning period. We do not impose an integrality restriction on this parameter, as we assume that a fractional portion of the voyage can be completed at the beginning of the following planning period.
p_j^v	Time at port j for vessels of type v .
w_r	Total port time for run r ; $w_r = \sum_{(i,j) \in \Delta_r} p_j^{v_r}$.
u_j	Cost of lifting a TEU at port j .
k_{od}	Total demand (in TEU) to the liner company for transport from o to d over the planning period.
q_{od}	Revenue from transport of one TEU from o to d .
\tilde{q}_{od}	Penalty for failing to transport one TEU from o to d .

Finally, the decision variables used in the MIP are:

X_{ijd}^r	Number of containers traveling to their final destination at port d along arc (i, j) on run r .
L_{id}^r	Number of containers to be transported to port d that are loaded onto run r at port i . This includes containers that are entering the network for the first time, as well as containers that are reloaded to run r after being transhipped from another run.

- U_{id}^r Number of containers traveling to port d which are unloaded from run r for transshipment at port i .
- W_d^r Number of containers traveling to port d reaching their final destination using run r .
- V_{od} Demand from port o to port d that is serviced by the liner company.
- O_{od} Demand from port o to port d that will not be serviced by the liner company.
- Y^r Number of vessels assigned to run r .

The MIP model is presented below, followed by an explanation of the objective function and all the constraints.

(VRD) Minimize

$$\begin{aligned}
Z_{MP} = & \sum_{r \in \Xi} Y^r f^{v_r} + \sum_{v \in \Upsilon} \tilde{f}^v \left(z^v - \sum_{r \in \Xi: v_r=v} Y^r \right) + \\
& \sum_{r \in \Xi} Y^r \cdot e \cdot m_r (g^{v_r s_r} b_r + h^{v_r} w_r) + \\
& \sum_{o \in \Pi} \sum_{d \in \Pi} \tilde{q}_{od} O_{od} - q_{od} V_{od} + \\
& \sum_{i \in \Pi} \sum_{r \in \Xi} u_i \cdot \left(W_i^r + \sum_{d \in \Pi} (L_{id}^r + U_{id}^r) \right)
\end{aligned} \tag{1}$$

subject to:

$$\sum_{j:(j,n) \in \Delta_r} X_{jnd}^r + L_{nd}^r = \sum_{j:(n,j) \in \Delta_r} X_{njd}^r + U_{nd}^r \quad \forall n, d \in \Pi \quad n \neq d \quad r \in \Xi \tag{2}$$

$$\sum_{j:(j,n) \in \Delta_r} X_{jnn}^r = W_n^r \quad \forall n \in \Pi \quad r \in \Xi \tag{3}$$

$$\sum_{n \in \Pi} O_{nd} + \sum_{r \in \Xi} W_d^r = \sum_{n \in \Pi} k_{nd} \quad \forall d \in \Pi \tag{4}$$

$$\sum_{r \in \Xi} L_{nd}^r = \sum_{r \in \Xi} U_{nd}^r + V_{nd} \quad \forall n, d \in \Pi \quad n \neq d \tag{5}$$

$$O_{nd} + V_{nd} = k_{nd} \quad \forall n, d \in \Pi \tag{6}$$

$$\sum_{d \in \Pi} X_{ijd}^r \leq c^{v_r} \cdot Y^r m_r \quad \forall r \in \Xi \quad (i, j) \in \Delta_r \tag{7}$$

$$\sum_{r \in \Xi: v_r = v} Y^r \leq z^v \quad \forall v \in \Upsilon \quad (8)$$

$$X_{ijd}^r, L_{id}^r, U_{id}^r, W_d^r \in \mathbb{R}^+ \quad \forall r \in \Xi \quad (i, j) \in \Delta_r \quad d \in \Pi \quad (9)$$

$$O_{nd}, V_{nd} \in \mathbb{R}^+ \quad \forall n, d \in \Pi \quad (10)$$

$$Y^r \in \mathbb{Z}^+ \quad \forall r \in \Xi \quad (11)$$

The first line on the objective function captures the fixed costs of operating the fleet, including laying up vessels that are not assigned to any run. The second line in the objective function obtains the fuel costs for all active vessels, which includes consumption both during steaming and while idling at ports. The third line in the objective function obtains the total revenues from transporting units as requested, and the penalties incurred when rejecting carriage requests. The final term in the objective function computes charges from loading and unloading containers, both at their origin and destination, and at transshipment points.

Constraints (2) balance the flow of containers loaded and unloaded from each run at points other than their final destination. Every container not destined for the port in question must either continue along in the same run or be unloaded for transshipment to a different run. Constraints (3) represent the flow of containers that were accepted for carriage arriving at their final destination. Every container arriving at its final destination is unloaded and grounded.

Adding the variables O_{od} to the model allows us to obtain a feasible solution for any fleet configuration. When there is not a sufficient number of vessels to transport all the demand, or when it is not economically convenient to transport all containers, the variables O_{od} are set to a positive value, indicating that some traffic demand has been forfeited. Constraints (4) tally traffic arriving at each port. This ensures that the amount of forfeited traffic, plus the traffic actually carried to the port, add up to the manifest demand for traffic into each port.

Constraints (5) represent the flow of containers originating and transhipped at each port. The total number of containers leaving the port must equal the traffic that enters the network from the hinterland of this port, plus any containers that were previously unloaded

for transshipment. Constraints (6) are analogous to constraints (4), but tally traffic from the perspective of the originating hinterland.

Constraints (7) impose restrictions on the total number of containers that can be transported by each run on each one of its arcs. The total that can be transported is limited by the number of vessels assigned to the run, the capacity of those vessels, and the number of trips the vessels perform over the planning horizon.

Constraints (8) ensure that the number of vessels that are deployed does not exceed the available vessels of each type.

Finally, constraints (9), (10), and (11) impose non-negativity and integrality restrictions on the corresponding variables.

2.3 Solution technique

Realistic instances of problem VRD pose a challenging computational problem. The difficulty in solving this problem centers on the presence of variables and constraints indexed over the set Ξ . In problems of relevant size, the number of possible runs (more specifically, the number of possible routes) will be so high that it may not be practical to list them, much less include them in a MIP. Secondly, given that larger vessels are more cost effective on a per-TEU basis, the linear programming relaxations of problem VRD will favor the use of *fractions* of the largest vessel available, and will not use any smaller vessels. These relaxations will therefore constitute very poor approximations of actual solutions, which detracts from the efficiency of traditional solution techniques such as branch and bound.

In order to solve problem VRD, we begin by noting that, if we fix the values of all variables Y^r to feasible values, we can eliminate constraints (8) and (11) from problem VRD. Additionally, by fixing the values of all variables Y^r , we fix the right-hand side of constraints (7). With these changes, problem VRD becomes a pure multicommodity flow problem (MCFP) (Ahuja, Magnanti, and Orlin, 1993). State of the art interior-point optimization codes can quickly solve very large instances of problem MCFP.

The preceding discussion suggests a two-tier solution approach whereby, at each iteration t , a set of runs $\Xi_{(t)}$ and the corresponding vessel deployment $Y_{(t)}^r$ are established at the higher

tier using tabu search (Glover and Laguna, 1997). The tabu mechanism identifies each run by concatenating, in alphabetical order, the names of the ports that are visited by the run². A run that has been recently modified can't be modified again as long as it appears in the tabu list. A run is considered to be modified by the following operations: creation, deletion, change in vessel type, change in vessel speed.

Using tabu search to control variables Y^r allows us to reinterpret the latter as parameters to the lower tier, problem $\text{MCFP}(\Xi_{(t)}, Y_{(t)}^r)$. Using information obtained from the solution of each problem $\text{MCFP}(\Xi_{(t)}, Y_{(t)}^r)$, we can then obtain an improved set of runs $\Xi_{(t+1)}$ and vessel deployments $Y_{(t+1)}^r$ at the higher tier. Figure (1) provides a schematic representation of the two-tier solution approach.

For the purposes of the case study presented in Section 2.4, we obtain an initial solution $\Xi_{(0)}$ and $Y_{(0)}^r$ by assigning small vessels to simple loops in each region $\lambda \in \Lambda$. We then connect the intra-region loops using large vessels on long-haul routes. In practice, it would be advantageous to set $\Xi_{(0)}$ and $Y_{(0)}^r$ to the actual configuration of the liner company, which would provide an initial solution of much higher quality.

At each iteration t of the algorithm, we solve problem $\text{MCFP}(\Xi_{(t)}, Y_{(t)}^r)$ and carefully analyze two sets of results derived from the solution: the load factors on each arc, and the shadow prices on constraints (7).

The total flow of containers using run r on arc (i, j) is given by $\sum_{d \in \Pi} X_{ijd}^r$ – the utilized capacity on arc (i, j) . Similarly, the installed capacity of run r on each one of its constituent arcs is given by $c^{vr} \cdot Y^r \cdot m_r$. By observing the ratio between the used capacity and the installed capacity, we can determine which runs are over- or under-utilized. Underutilized runs are good candidates for a reduction in the number of vessels deployed to them or for elimination from the set $\Xi_{(t+1)}$. Alternatively, we can decrease the size or speed of some of the vessels employed in that run. Overutilized runs are good candidates for an increase in the number, size, or speed of vessels deployed to them.

The shadow prices ρ_{ij}^r associated with each constraint (7) indicate the potential impact

²By sorting the ports of the restricted run in alphabetical order, we extend the tabu restriction to any runs that consist of a permutation of the same ports.

to the objective function from a unit increase on the right-hand side of the constraint. If we were to add a new vessel to run r , the right-hand side of every constraint (7) associated to the links incident upon r would increase by $c^{v_r} \cdot m_r$. The potential benefits derived from this additional capacity are approximated by $\sum_{(i,j) \in \Delta_r} \rho_{ij}^r \cdot c^{v_r} \cdot m_r$. We can therefore estimate the overall benefit of adding this vessel to the run by comparing the latter quantity to the costs required to deploy a vessel v_r .

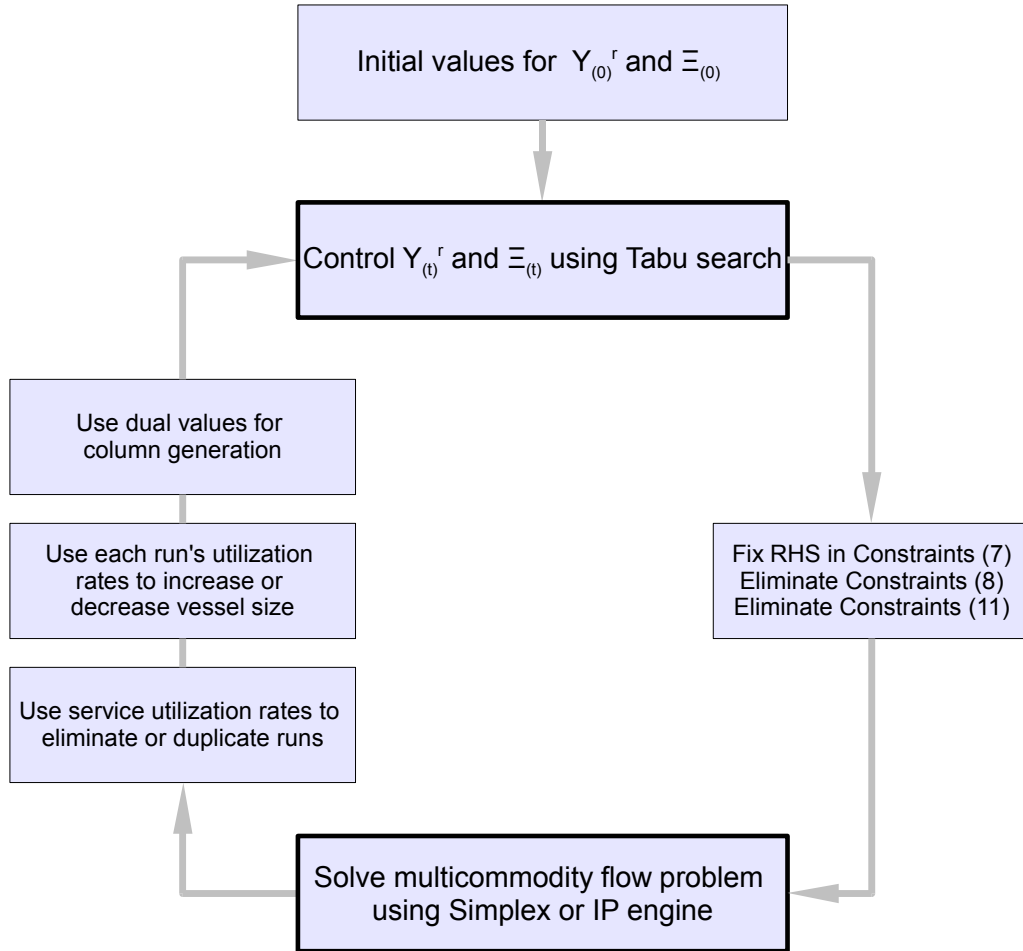


Figure 1: Schematic representation of the two-tier solution approach.

In fact, we can use a similar idea to estimate the benefit of adding an altogether new run r' . In order to do this, we imagine adding a dummy run a to each problem MCFP. This dummy run has zero units of capacity along every arc in the network, but serves as a

placeholder for the shadow prices of arcs where no service currently exists. We then design the route $\Delta_{r'}$ of the new service, and estimate the benefits derived from the capacity of the new run by $\sum_{(i,j) \in \Delta_{r'}} \rho_{ij}^a \cdot c^{v_{r'}} \cdot m_{r'}$.

We use this method to generate a multitude of possibly beneficial new runs, in what is known as a column generation procedure. Formally, for each vessel type v and speed s , we seek to solve problem $AUX(vs)$, which determines the composition of a column with negative reduced cost in problem $MCFP(\Xi(t), Y_{(t)}^r)$. There are several difficulties that prevent us from addressing this problem directly. First, given that the route to be followed in run r' is not known, the value of parameter $m_{r'}$ (the number of round trips that a vessel will perform on this run) is not known either. Introducing this parameter as a decision variable in the problem would add an unwelcome non-linearity. We can circumvent this problem by looking for new columns that have negative reduced cost *per trip*. By changing the focus of the column generation procedure from total impact on the objective function to impact per trip, we do not exclude any valid columns.

The second difficulty follows from our intention to construct services that are loops. Unfortunately, this introduces the possibility that many of the columns generated by the model would contain disjoint subtours. The elimination of these subtours from the proposed solutions poses a significant computational challenge, akin to the elimination of subtours in the traveling salesman problem.

We propose a simple heuristic to discard solutions containing disjoint subtours. We start by setting a limit i^{vs} on the savings to be obtained from each newly proposed run that uses a vessel of type v steaming at speed s . Initially, we set $i^{vs} = -\infty$. After solving $AUX(vs)$, if the resulting solution does not contain subtours, we accept the candidate run. Otherwise, we set $i^{vs} = Z_{AUX(vs)} + \delta$, with $\delta \geq 0$, and solve the problem again. The effect of this adjustment is to prevent the most recent solution from appearing again. We proceed in this manner until a valid run is obtained, or until the magnitude of i^{vs} becomes very small, indicating that no worthwhile new runs using this vessel-speed combination can be found.

The auxiliary problem $AUX(vs)$ is now presented. We start by defining some additional

parameters:

- i^{vs} Lower bound on the financial benefit obtained from the addition of the new run.
- ρ_{ij} Dual value associated to constraint (7) for arc (i, j) in the most recent solution to problem MCFP. Where there is no run incident on arc (i, j) at iteration t , we use ρ_{ij}^a . Otherwise, we use $\min_{r \in \Xi_{(t)}: v_r=v, s_r=s} \rho_{ij}^r$, as several runs may be incident on arc (i, j) .
- m_{min} Minimum number of voyages required over the planning horizon.
- T Length of the planning horizon.

Problem AUX (vs) has the following decision variables:

- A_{ij}^{vs} Binary variable, indicates whether arc (i, j) forms part of the new run.
- μ^{vs} *Inverse* of the number of trips to be completed on the new run over the entire planning horizon.
- ω^{vs} Estimated per-trip cost of the new run on the objective function of problem MCFP($\Xi_{(t)}, Y_{(t)}^r$).

The auxiliary column generation model is then given by:

AUX(vs) Minimize

$$Z_{AUX(vs)} = \omega^{vs} \quad (12)$$

subject to:

$$\omega^{vs} = \mu^{vs}(f^v - \tilde{f}^v) + \sum_{(i,j) \in \Delta} A_{ij}^{vs} (e \cdot h^v p_j^v + e \cdot g^{vs} l_{ij}^v - \rho_{ij} c^v) \quad (13)$$

$$T \cdot \mu^{vs} = \sum_{(i,j) \in \Delta} A_{ij}^{vs} \left(p_j^v + \frac{l_{ij}^v}{s} \right) \quad (14)$$

$$\sum_{j \in \Delta} A_{jn}^{vs} = \sum_{j \in \Delta} A_{nj}^{vs} \quad \forall n \in \Pi \quad (15)$$

$$i^{vs} \leq \omega^{vs} \quad (16)$$

$$\mu^{vs} \leq \frac{1}{m_{min}} \quad (17)$$

$$A_{ij}^{vs} \in \{0, 1\} \quad \forall (i, j) \in \Delta \quad (18)$$

Constraints (13) and (14) serve to define the values of ω^{vs} and μ^{vs} . Constraints (15) require that each candidate route be composed of closed loops. For the purposes of the subtour elimination heuristic, constraint (16) imposes a limit on the benefit that can be obtained from the new service. Constraint (17) limits the duration of each voyage of the new service to comply with crewing requirements.

A key feature of this column generation scheme is that it has the potential to generate routes of any type: triangle services, pendulum services, butterfly services, conveyor belt services, and combinations thereof. It should be noted that problem $AUX(v, s)$ could be supplemented by purely heuristic or rules-based column generation methods, which can generally embody more sophisticated restrictions on the types of routes that are deemed acceptable.

2.4 Case study

We propose a study of the sensitivity of routing and deployment policies to bunker prices. This analysis is particularly relevant at present, when oil prices have surged to record prices (c.f. Figure 2), placing a tremendous burden on the transportation industry.



Figure 2: Crude oil prices 2007- 2008 (The Wall Street Journal, 2008)

Although the data we use in the case study are realistic, they do not describe any liner company in particular. We now comment briefly on the assumptions and methods used to generate the case data.

We created a short list of prototypical vessels based on the web site of Hapag-Lloyd (2007), which provides a catalog of vessel classes used by the company. For each vessel class, the web site lists the vessel’s capacity, design speed, and engine power at the design speed. We generated cost and fuel consumption profiles for each vessel using formulas and data provided by Alderton (2005). We estimated fuel consumption at the service speed based on the power generated by each vessel’s engine, and assuming a specific consumption of 125 grams per brake horsepower (bhp) per hour. We calculated fuel consumption at a lower speed (80% of the design speed) using the cube law:

$$Consumption_{80} = Consumption_{design} \cdot (speed_{80}/speed_{design})^3 \quad (19)$$

The main characteristics of the fleet used in the case study are listed in Table 1.

Vessel type	Cap. [TEU]	idle cons [t/dy]	Low Speed		Design Speed		Nb Avail
			Sp[kn]	Cons[t/dy]	Sp[kn]	Cons[t/dy]	
SuperPPmx	8,400	2.5	20.0	157	25.4	305	20
PostPmx	4,900	2.0	19.2	97	24.0	188	20
Panamax	3,200	1.7	16.8	59	21.6	114	20
feeder2	2,100	1.3	15.2	33	19.7	64	20
feeder1	1,000	1.0	14.8	26	18.5	51	20

Table 1: Available vessel types

We obtained a list of the top 120 container ports from Container Management Magazine (2007). We generated a matrix of optimal sailing distances between these ports using data from the National Imagery and Mapping Agency (NIMA, 2001). The source data lists distances from each port to the most important ports in its vicinity, as well as distances to major routing junction points (e.g. the Panama Canal, the Straits of Malacca, etc.). We ran a many-to-many shortest path algorithm to determine the best routes between each pair of ports in our top 120 list. We disabled several routes that would require ice strengthening, or that would not meet the draft requirements of large container vessels.

We generated the port-to-port transportation demand matrix using a logit model. We started with a total yearly demand for transportation of 5 million TEU, corresponding to a fairly large liner company. In order to create a realistic geographic distribution of demand, we allocated this demand among countries in proportion to each country's gross domestic product (GDP). Next, we allocated the demand for each country among that country's top ports in proportion to each port's ranking in the list of the top 120 container ports. Finally, we determined the demand for transportation between each pair of ports using a logit model:

$$k_{od} = k_o \cdot \exp(-l_{od}^{Pmx}) / \sum_{j \in \Pi} \exp(-l_{oj}^{Pmx}), \quad \forall o, d \in \Pi \quad (20)$$

where k_o is the total demand for transportation generated in the hinterland of port o , and l_{ij}^{Pmx} represents the sailing distance between ports i and j for a Panamax vessel. Figure 3 depicts the demand for transportation that is generated by the hinterland of each port in this model. In the figure, the size of the circle that indicates each port is proportional to k_o .

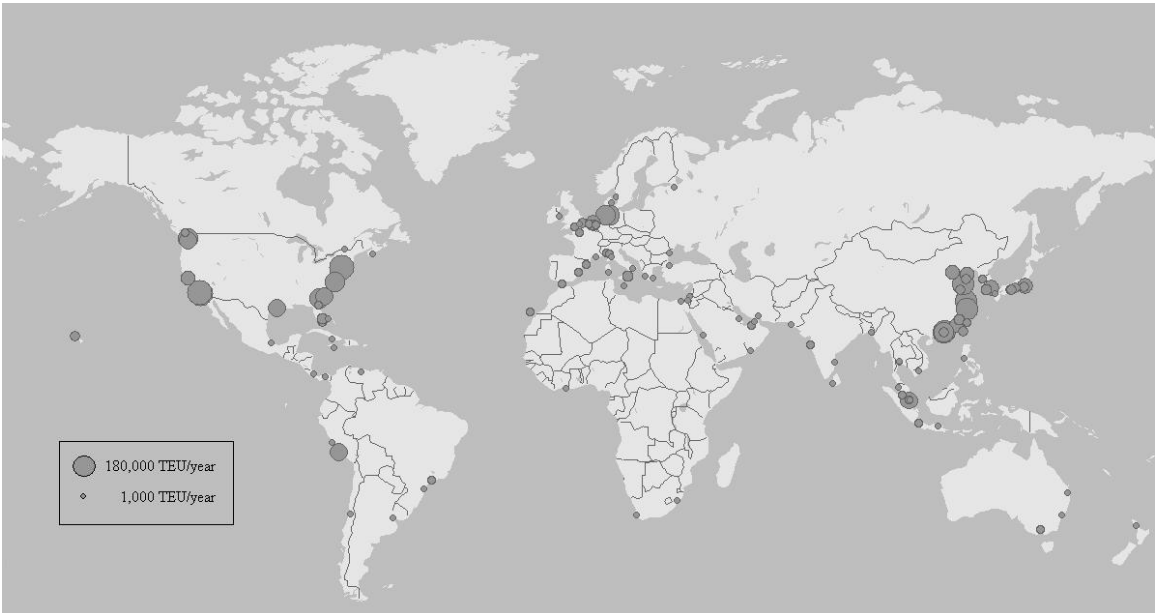


Figure 3: Transportation demand at each port

We implemented the multicommodity and column generation portions of the algorithm using AMPL 10 (Fourer, Gay, and Kernighan, 2003) and CPLEX 10.0 (ILOG S.A., 2006). The master algorithm, which controls the tabu metaheuristic and the calls to the AMPL/CPLEX codes, was implemented in C++. All experiments were carried out on a 2.1 MHz Pentium computer with 2GB of RAM.

First, we ran a base scenario with bunker prices at US\$400 per ton and very high penalties for rejecting traffic. The algorithm produced 85 services and deployed 91 vessels. At most three vessels were deployed on any service. The services that used multiple vessels generally employed vessels of different types. The algorithm used all available Feeder1, Feeder2, Panamax, and PostPanamax vessels. Overall, fewer than 12% of the vessels were deployed at their design speed. In this solution, a total of 2.6 million TEU-miles was provided by the fleet.

The second scenario was identical to the first, except for a higher bunker price of US\$600 per ton. In this scenario, the algorithm deployed 87 vessels on 83 services. Relative to the base case, four services were cancelled and 13 were modified to cover different routes. Furthermore, two Feeder2 vessels were redeployed at the lower speed. Three PostPanamax and one SuperPostPanamax vessels were docked. The total TEU-miles provided by the fleet went down by 10%, while the number of rehandled containers increased by 9.5%. The amount of fuel consumed decreased by 7% due to the route adjustments.

In the third scenario, the cost of fuel was set at US\$800 per ton. As before, the algorithm suggested the deployment of 87 vessels on 83 services. However, relative to the second scenario, five routes were changed. Furthermore, one SuperPostPanamax vessel was docked, while one PostPanamax vessel was placed in service. The number of rehandles increased slightly, while the total TEU-miles provided by the fleet decreased slightly. The total fuel consumed decreased by 3%.

In all three scenarios, over 70% of ports received a vessel at least once a week, and 90% of ports received a vessel at least every ten days. However, some important ports were served infrequently. It is therefore likely that some additional level-of-service constraints would be required in the proposed model.

2.5 Conclusions

We have presented a mathematical programming formulation for the joint routing and deployment of a fleet of container vessels. This problem is quite complex from the mathematical, business, and computational perspectives, and there exist abundant opportunities for further research and enhancements. We outline the most pressing questions below.

The methodology proposed here combines exact mathematical programming algorithms and a meta-heuristic guidance mechanism. As is generally the case, the heuristic component offers some speed enhancements, but does not guarantee termination at an optimal point. Rather, the algorithm may terminate at a local optimum. Vanderbeck and Wolsey (1996) propose an exact algorithm for integer formulations that require column generation. The implementation of such an algorithm for problem VRD is an interesting direction for further research. However, given that the branch and bound methods generally require column generation at each node, it can be anticipated that the computational requirements of this approach would be substantial.

The implementation presented here uses a generic interior point algorithm to solve the multicommodity flow subproblems. For instances of relevant dimensions, the interior point algorithm appears superior to primal and dual simplex codes. This is likely due to the sparsity in the columns of problem MCFP. It remains to be seen whether this advantage is sufficient to overcome the benefits of an advanced basis initialization in the simplex implementations. Additionally, the use of specially tailored interior point algorithms to solve the multicommodity flow subproblems could provide a significant increase in performance (Castro, 2003).

In the proposed model, a major source of computational complexity arises from the joint treatment of fleet deployment and route design. With few exceptions, research in the maritime sector has thus far treated these two problems separately. A natural direction for future research consists of assessing the benefits of the joint treatment when compared to the more common, and significantly more tractable, approach.

One final line of work, from the mathematical perspective, is the derivation of lower bounds for problem VRD, which would allow us to retain the heuristic approach, while

providing an estimate of the optimality gap.

From the business perspective, we should stress that the proposed model is best employed in the context of a specific liner company. Many of the actual business aspects of the vessel routing and deployment problem can be readily represented using the proposed model. However, each company will have specific guidelines for the design of its services. The models and algorithms proposed here will need to be tailored accordingly.

Transshipment points in the model serve to decouple the transit times of vessels and containers. This generates an opportunity to devise, and bring to market, service levels differentiated by transit times, a step in the direction of revenue management (Ting and Tzeng, 2004). On the other hand, this generates the operational challenge of ensuring that the promised transit times can be achieved using the vessel deployments that result from the model proposed here. The formulation of such a model, as well as the inclusion of more precise measures of level of service, are interesting directions for future research.

The proposed model constitutes a valuable tool for performing what-if analyses. One such exercise was presented as a case study, where the sensitivity of vessel routing patterns to bunker price fluctuations was studied. The model could be readily employed to analyze the impact of entering new markets, vessel acquisitions, fleet mergers, variations in demand for transportation, and delays at specific ports, among others.

Chapter 3

A Lagrangean Relaxation Approach to the Vessel Planning Problem

Due to the high cost of stevedoring services and equipment, competition amongst terminals, and the high congestion at many major ports (White and Earnest, 2005), maritime terminal operations must be carried out with the utmost efficiency (Alderton, 2005). One of the most important duties of the terminal staff is directing the timely loading of outbound containers on to the ship. Although most modern terminals employ information systems to assist the ship planners in the generation of the vessel loading plan, optimization algorithms are still not in widespread use in this application.

The existing literature provides several exact and approximate algorithms that permit greater automation of ship load planning (Böse et al., 2000; Steenken et al., 2001; Kim et al., 2004; Imai et al., 2006). However, most previous research has focused on terminals that utilize rubber tired gantry cranes (RTGs), rail mounted gantry cranes (RMGs), and straddle carriers (SCs) to store and retrieve containers (Steenken et al., 2004; Stahlbock and Voß, 2007). The present chapter focuses on container terminals that use reach stackers, which have received scarce attention in the scientific literature.

In terminals that utilize SCs, containers are stacked in single rows, three or four tiers high (Montfort et al., 2001). Adjacent rows are separated by corridors where the SCs travel. One drawback of this layout is that the travel paths take up a significant portion of the

yard’s valuable floor space. On the other hand, any outbound container in the yard can be reached after removing at most three containers.

In terminals that utilize reach stackers, RTGs, or RMGs, containers are stored in dense blocks, up to five tiers high, several dozen containers wide, and six to twelve containers deep. RTGs and RMGs access the stacks from above, so the potential number of rehandles is again limited. On the other hand, reach stackers access containers from the sides of the block, so it may be necessary to remove and then restack (i.e. rehandle) a large number of containers to reach a needed container.

Container rehandling is costly to the terminal, as it reduces operational efficiency and results in extended ship berthing times – a key performance indicator for terminals. Furthermore, many stevedoring unions charge a fee per container moved, whether the container is loaded on the ship or not. Clearly, it is essential to sequence the loading of outbound containers in such a way as to minimize rehandles.

Generating the loading plan by hand is tedious and extremely time consuming. The central contribution of this chapter is an efficient algorithm for automated vessel planning in container terminals employing reach stackers.

The remainder of the chapter is organized as follows. Section 3.1 provides a detailed description of the ship load planning problem. Section 3.2 presents a review of relevant literature in this area. In Section 3.3 the vessel planning problem is formulated as a MIP model and decomposed into three simpler problems using Lagrangean relaxation. Section 3.3.2 presents an exact algorithm to resolve the vessel planning problem as well as several implementation issues that are essential to the performance of the algorithm. Section 3.5 provides results from computational experiments using data from a Spanish container terminal. Conclusions are presented in Section 3.6.

3.1 The load planning process

The yard of a container terminal is organized using four hierarchical levels: blocks, bays, lanes, and tiers (Zhang et al., 2002). A block has a large rectangular footprint, and is composed of several bays (distributed along the long axis of the block), which are in turn

composed of stacks. Loaded containers are stacked up to five tiers high. The exact location of a container in the yard is thus given by specifying values for the four aforementioned fields.

Each block in the yard is designated as either an import or export block. Import blocks accommodate containers that have arrived at the terminal and are awaiting pickup by a truck or train. The process that assigns import containers to yard locations is known as decking. The constraints that govern the decking problem differ from those that govern the loading of containers on to the vessel, and these two problems are typically solved separately. The decking problem is outside of the scope of the present research.

At each step of the loading process, the reach stacker operator receives instructions (either through a computer terminal, or through a two-way radio) indicating the yard location of the next container that is to be loaded. The reach stacker removes this container from the stacks and loads it on a flat-bed truck. The truck then proceeds to the quayside, where a quay crane removes the container from the truck and immediately loads it on the vessel.

Loading the export containers requires extensive planning to ensure an efficient operation. When generating the load plan, the terminal staff attempts to expedite the loading operation while satisfying a number of constraints, such as: compliance with the stowage plan, preservation of ship stability, the availability of stevedoring crews and equipment, the spatial distribution of the containers in the yard, visibility restrictions of the crane operators, the safety and efficiency of the latching crew¹, and interference between multiple stevedoring teams.

Prior to the arrival of the ship at the terminal, the ship's operator transmits two sets of data to the terminal staff: the ship's bay plan and the stowage plan. The bay plan describes the nature and position of each container currently on board the vessel, and is typically transmitted using the EDIFACT BAPLIE (SMDG, 2001) standard. The stowage plan indicates the ship slots that are to be occupied by the export containers, and the *class* of containers each slot can accommodate. The stowage plan is typically transmitted using

¹Members of the crew secure the containers that have been loaded using steel rods.

the EDIFACT MOVINS (SMDG, 2000) standard. The container type², destination port, hazardous materials IMO³ designation, and weight together constitute the container class (Kim et al., 2004). The yard will likely have several export containers of a given class. Therefore, the terminal staff must decide which specific container will be loaded in each slot as well as the sequence in which the containers will be loaded, a process known as vessel planning.

The vessel planning process will typically consist of the following steps. First, the staff must decide how many stevedoring teams will work on the ship in question. The labor laws and agreements relating to stevedoring practice are complex and vary between localities. The staff at each terminal must be aware of such regulations, as they generally stipulate the number of hours that a team can work in a shift; the number of workers in each team; whether and when a team can be assigned to work sequentially on multiple ships; compensation to the stevedores (including charges for idling, per container moved, and surcharges for night or holiday shifts); and the composition of each team (typically a crane operator, a reach stacker operator, one or two truck drivers, a foreman, and one or more ladders on board the ship). The number of teams assigned to the ship also depends on the size of the ship, the number of containers to load, and the level of activity at the terminal.

Once a decision has been made on the number of teams to employ, the ship's holds must be allocated amongst them. If all teams are expected to work for the entire shift on the particular vessel, the goal is to balance the workload of all the teams. Alternatively, the staff may attempt to reduce the workload of one team, in hopes of later reassigning it to a different vessel. After the holds have been assigned to the teams, the planner decides the direction of loading for each hold (fore to aft, or vice versa). In doing so, the planner must ensure that all quay cranes are able to operate simultaneously without interfering with each other.

²Commonly used container types are: DC (standard size dry cargo), HC (high cube), OT (open top), RF (refrigerated), HR (refrigerated high cube), and TA (tank), amongst others.

³IMO, the International Maritime Organization, an agency of the United Nations, aims to improve the safety and security of international shipping (International Maritime Organization, 2006).

The quay cranes can load the containers on to each hold using one of several *disciplines*. As a general rule, and to ensure adequate visibility for the crane operator, containers are loaded starting from the water side, and moving towards the quay side. Within the limits of this restriction, the containers can be loaded vertically by columns, or horizontally by layers. In either case, once the loading discipline has been chosen, the order in which the slots of a given hold are loaded is mostly fixed (Steenken et al., 2001). Occasionally, the planner will allow small deviations from the chosen discipline. However, for the remainder of the dissertation, we assume that the stowage plan and the loading discipline employed by each crane uniquely determine a sequence of generic containers to be loaded on the ship.

The role of the planner from this point on is to assign a specific container to each generic slot in the vessel, which (in light of the previous assumptions) determines the order in which the containers are to be retrieved from the yard. The planner must also specify the *approach side* when removing each container. In most cases, a container can be reached from either of the two aisles adjacent to the block, which are often referred to as water-side and land-side. Starting from one such side may result in a lower number of rehandles for the current step and may reposition the reach stacker favorably for subsequent operations. In the forthcoming, the focus will be on the approach *transitions*, which determine how each reach-stacker is repositioned between two steps of the loading process.

In sequencing the loading of containers, the planner attempts to optimize three competing objectives:

- Rehandling of containers. When the reach stacker operator is asked to access a container that is covered by others, he must first remove all the containers that are impeding access to the target container. After removing the target container and loading it on to the truck that delivers it quayside, the reach stacker operator must generally restack the containers that were initially removed. This task is complicated because the reach stacker has limited room to maneuver in the aisles between blocks. Storing containers in the aisles, even temporarily, is not practical.
- Reach stacker displacements. Movement of the reach stacker around the yard must be

minimized, as this equipment travels slowly, and is subject to costly wear and tear.

- Vessel instability. The planner must also be mindful of the correct distribution of weight on board, which affects the ship’s trim and stability. Here, it must be stressed that there is no theoretical objection to a heavier container being placed on top of a lighter one, as long as the overall stability of the vessel remains acceptable. Nevertheless, most planners place great emphasis in stowing heavier containers at the bottom of the holds.

At present, the vessel planning process has not been fully automated at many terminals (this is particularly true for reach stacker terminals, which are typically smaller and not as technologically advanced as RMG or RTG terminals). In most cases, the process is performed with the aid of a terminal operating system (TOS), which mainly aids in the visualization of the yard and ship holds, without adding substantial optimization-based decision support. In less sophisticated terminals, vessel planning entails working on printed copies of the preliminary stowage plan and a list of available containers obtained from a central database. Groups of containers are highlighted in different colors to indicate their load sequence. Manually generating the plan for a ship loading 500 containers may take between six and eight hours. There is no generally accepted method to judge the goodness of a given load plan, and the measures of quality involved are given different priorities by different terminals and planners.

3.2 Literature review

Steenken et al. (2004) provide an excellent classification of processes in container terminals that are amenable to optimization techniques. They provide a thorough review of the literature in this field, with over two hundred references up to 2004. The authors provide introductory descriptions of the cranes, ground transportation equipment, and information systems used in modern container terminals. The authors then describe the berth allocation, stowage planning, crane split, quayside transport, yardside transport, and container stacking problems. Vis and de Koster (2003) also provide an overview of container terminal operations and a comprehensive review of the literature in the field.

Steenken et al. (2001) propose exact and heuristic methods to address the combined stowage planning and ship loading problem. Their paper focuses on terminals utilizing straddle carriers for stacking and ground transport. In particular, they present experimental results from the Port of Hamburg. Their algorithms perform a simple crane split, similar to the one proposed in the present chapter. Their MIP formulation accounts for the different transportation times required by containers located in different sectors of the yard, and attempts to minimize lateness with respect to the quay crane's load cycle. A Δ -replan technique is proposed to address the common problem of outbound containers that arrive at the terminal after the load operation has begun. This technique limits the load planning horizon to a fixed number of containers ahead of the current operation, thus creating a partial load plan using containers that are already in the yard.

Kim (1997) proposes exact and approximate formulas to determine the number of rehandles on import blocks. The results are applicable only to blocks operated using transtainers. The approximate formula is based on a linear regression of the exact result against a range of different block layouts.

Bish et al. (2005) provide simple greedy heuristics to address the vehicle dispatching problem given a complete ship loading plan. Their algorithms attempt to minimize the ship berthing time. The authors assume that a fixed number of vehicles has been assigned to serve each quay crane. In the single-crane case, they demonstrate the optimality of greedy and reverse-greedy strategies for the pure unloading and loading problems, respectively. For the multiple crane scenario, they provide a minor modification to the greedy algorithm that results in solutions of remarkably high quality. The authors compare the results of their heuristics to the solutions derived from a MIP formulation.

Kim et al. (2004) propose a beam search algorithm for load sequencing in container terminals. Their paper focuses on terminals that operate with transtainers. The algorithm considers many important side constraints, such as the total height and weight of containers on board, and interference between multiple yard cranes. They provide a mathematical formulation of the problem, but the resulting model is highly non-linear on integer and linear variables. The proposed heuristic operates at two levels of granularity. The first level

selects a module in the yard from which a series of containers will be extracted. The second level determines the order in which the containers from a given module will be loaded.

Kozan (2000) presents a network model to minimize the total throughput time in a multimodal container terminal. The model is used to analyze the impact of the quantity of equipment in operation on throughput time. For a case study in the port of Brisbane, Australia, the author finds that an increase in the number of quay cranes has the largest impact on terminal efficiency.

Vis and Harika (2004) created a discrete event simulation to study the effectiveness of different vehicle types in a container terminal. Their study focuses on automated guided vehicles and automated lift vehicles. The simulation model is used to determine the number of vehicles required to provide the desired level of service. A sensitivity analysis examines the impact of different vehicle speeds, quay-side buffer areas, and quay crane speeds on the results.

3.3 Mathematical model

We make the following assumptions in developing the mathematical formulation of the vessel planning problem.

- The terminal has a sufficient number of containers of each class to fulfill the requirements of the stowage plan. Furthermore, if the terminal has an excess of containers of a given class, the planner has determined which containers will not sail.
- All stevedoring teams working on a vessel will remove containers from the yard at a similar pace. The model therefore refers to a sequence of discrete time slots, with each stevedoring team processing one container per time slot.
- All containers to be loaded have a known position in the yard. In practice, a substantial number of containers will arrive in the yard at the last moment, as the vessel is being loaded. Nevertheless, these so-called *to come* containers will typically be placed in designated buffer zones, so an approximate position can be established.

- All decisions regarding the number of teams to employ, assignment of vessel holds to each team, and loading discipline for each hold are made by the planner prior to launching the proposed algorithm. Therefore, the algorithm receives a unique sequence of slots (with their corresponding class) to be filled by each reach stacker.

3.3.1 MIP formulation

In this section, the ship load planning problem is formulated as a MIP. The following sets are used in the formulation:

Ω	Set of all reach stackers working on the vessel.
Ξ	Set of all containers to be loaded.
Π_r	Set of all vessel slots to be filled by reach stacker r . $ \Pi_r = P_r$.
Π'_r	Set of all but the first slot to be filled by reach stacker r .
Γ	Reach stacker approach transitions. $\Gamma = 1, 2, 3, 4$. By convention, $g = 1$ denotes a water-side approach at step $p - 1$ and a water-side approach at step p . Similarly, $g = 2$ denotes a water-land transition, $g = 3$ denotes a land-water transition, and $g = 4$ denotes a land-land transition.
Π_{rj}	Set of vessel slots to be filled by reach stacker r compatible with container j .
Ξ_{rp}	Set of all containers that are compatible with slot p , $p \in \Pi_r$.

The following parameters represent the known data for the problem:

w_i	Weight of container i .
l_{rp}	Instability penalty per unit of weight placed in slot p , $p \in \Pi_r$.
d_{ijg}	Ground distance between containers i and j when transitioning type is g .
b_{ijg}	Number of rehandles initially required to remove container j when starting from the position of container i and transitioning type is g .
v_{jg}	Number of containers required to access container j when it is the first container in the sequence, and transitioning type is g .

ε_{kijg}	Binary, container k initially blocks access from container i to container j when transitioning type is g .
α_s	Objective function coefficient for container rehandling disutility.
α_d	Objective function coefficient for ground travel of the reach stackers.
α_w	Objective function coefficient for weight distribution on board the vessel.
M	A suitably large number.

Finally, the decision variables used in the MIP are:

X_{rpj}	Binary, equal to one iff reach stacker r loads container j into slot p , $p \in \Pi_r$.
A_{rpg}	Binary, equal to one iff approach transition of reach stacker r is g at load step p , $p \in \Pi_r$.
R_{rp}	Number of containers cleared by reach stacker r when filling slot p , $p \in \Pi_r$.
D_{rp}	Distance traveled by reach stacker r when loading container for slot p , $p \in \Pi_r$.
W_{rp}	Instability disutility for slot p , $p \in \Pi_r$.

The MIP model is presented below, followed by an explanation of the constraints.

(VPMIP) Minimize

$$Z_{MIP} = \alpha_s \sum_{r \in \Omega} \sum_{p \in \Pi_r} R_{rp} + \alpha_d \sum_{r \in \Omega} \sum_{p \in \Pi_r} D_{rp} + \alpha_w \sum_{r \in \Omega} \sum_{p \in \Pi_r} W_{rp} \quad (21)$$

subject to:

$$\sum_{j \in \Xi_{rp}} X_{rpj} = 1 \quad \forall r \in \Omega, \forall p \in \Pi_r \quad (22)$$

$$\sum_{r \in \Omega} \sum_{p \in \Pi_{rj}} X_{rpj} = 1 \quad \forall j \in \Xi \quad (23)$$

$$\begin{aligned} R_{rp} \geq & b_{ijg} - \sum_{q \in \Omega} \sum_{s=1}^{p-1} \sum_{k \in \Xi_{qs}} X_{qsk} \varepsilon_{kijg} \\ & - M(3 - A_{rpg} - X_{rpj} - X_{r,p-1,i}) \quad \forall r \in \Omega, \forall p \in \Pi'_r, \forall i \in \Xi_{r,p-1} \\ & \quad \quad \quad \forall j \in \Xi_{rp}, \forall g \in \Gamma \end{aligned} \quad (24)$$

$$R_{r1} \geq \sum_{j \in \Xi_{r1}} v_{jg} X_{r1j} - M(1 - A_{r1g}) \quad \forall r \in \Omega, \forall g \in \Gamma \quad (25)$$

$$D_{rp} \geq d_{ijg} - M(3 - A_{rpg} - X_{rpj} - X_{r,p-1,i}) \quad \forall r \in \Omega, \forall p \in \Pi'_r, \forall i \in \Xi_{r,p-1} \\ \forall j \in \Xi_{rp}, \forall g \in \Gamma \quad (26)$$

$$W_{rp} = \sum_{j \in \Xi_{rp}} l_{rp} w_j X_{rpj} \quad \forall r \in \Omega, \forall p \in \Pi_r \quad (27)$$

$$A_{r,p-1,1} + A_{r,p-1,3} - A_{rp2} - A_{rp1} = 0 \quad \forall r \in \Omega, \forall p \in \Pi'_r \quad (28)$$

$$A_{r,p-1,2} + A_{r,p-1,4} - A_{rp3} - A_{rp4} = 0 \quad \forall r \in \Omega, \forall p \in \Pi'_r \quad (29)$$

$$\sum_{g=1}^4 A_{r,0,g} = 1 \quad \forall r \in \Omega \quad (30)$$

$$\sum_{g=1}^4 A_{r,P_r+1,g} = 1 \quad \forall r \in \Omega \quad (31)$$

$$X_{rpj}, A_{rpg} \in \{0, 1\} \quad \forall r \in \Omega, \forall p \in \Pi_r, \forall j \in \Xi_{rp}, \forall g \in \Gamma \quad (32)$$

$$R_{rp}, D_{rp}, W_{rp} \geq 0 \quad \forall r \in \Omega, \forall p \in \Pi_r \quad (33)$$

Constraints (22) and (23) require that each container be assigned to exactly one vessel slot. Constraint (24) serves to compute the number of rehandles at step p . The constraint starts with b_{ijg} , the number of rehandles required to reach the container being loaded in step p when no containers have been removed from the yard. The triple summation serves to adjust the initial rehandle count downward, according to the number of containers that have been already loaded, and that were previously an obstacle to reaching j , the target container. Constraint (25), a specialized version of constraint (24), serves to compute the number of shifts at the first step in the loading sequence. Constraint (26) serves to compute the total ground distance traveled during step p . In constraints (24), (25), and (26), the M coefficient (often called *big-M*) is chosen so that each constraint becomes redundant unless the quantity being multiplied by M is zero. Constraint (27) serves to compute the weight distribution penalty generated by the p^{th} container loaded by reach stacker r . Constraints (28), (29), (30), and (31) ensure the consistency of the attack transitions A_{rpg} . Although other formulations of these four constraints are possible, this representation is particularly convenient because the associated portion of the constraint matrix is totally unimodular. In fact, it will be shown that these constraints can be directly translated into a shortest

path problem. Constraints (32) and (33) address the integrality and non-negativity of the respective decision variables.

In order to tighten the formulation, we replace parameter M for b_{ijg} and d_{ijg} in constraints (24) and (26), respectively. We also replace M by $\hat{v}_g = \max_j v_{jg}$ in constraints (25). Finally, we remove constraints (27) by adding the weight distribution penalty terms W_{rp} directly to the objective function.

3.3.2 Lagrangean relaxation

Given the presence of binary variables in the vessel planning problem, we attempt a solution using branch and bound. Unfortunately, constraints (24), (25), and (26) are not well suited for a solution based on a linear programming (LP) relaxation of VPMIP – fractional values of X_{rpj} or A_{rpg} are very likely to produce infeasible solutions. On the other hand, given the special structure of the vessel planning problem, good primal solutions can be obtained by using Lagrangean relaxation. We start by noting that, in the absence of the complicating constraints (24), (25), and (26), problem VPMIP can be separated into three independent subproblems, each of which can be solved using very efficient algorithms:

- A bipartite assignment problem on variables X_{rpj}
- A shortest path problem on variables A_{rpg} , and
- A trivial problem on variables R_{rp} and D_{rp}

A natural approach to obtain the Lagrangean relaxation is then to dualize constraints (24), (25), and (26) by associating to them Lagrange multipliers μ , η , and λ , respectively. This results in problem VPD, below.

(VPD) Minimize

$$\begin{aligned}
 Z_{LG}(\mu, \eta, \lambda) = & \alpha_s \sum_{r \in \Omega} \sum_{p \in \Pi_r} R_{rp} + \alpha_d \sum_{r \in \Omega} \sum_{p \in \Pi_r} D_{rp} + \alpha_w \sum_{r \in \Omega} \sum_{p \in \Pi_r} \sum_{j \in \Xi_{rp}} l_{rp} w_j X_{rpj} \\
 & + \sum_{r \in \Omega} \sum_{p \in \Pi'_r} \sum_{i \in \Xi_{r,p-1}} \sum_{j \in \Xi_{rp}} \sum_{g \in \Gamma} \mu_{rpijg} b_{ijg} (A_{rpg} + X_{rpj} + X_{r,p-1,i} - 2)
 \end{aligned}$$

$$\begin{aligned}
& - \sum_{r \in \Omega} \sum_{p \in \Pi'_r} \sum_{i \in \Xi_{r,p-1}} \sum_{j \in \Xi_{rp}} \sum_{g \in \Gamma} \mu_{rpig} (R_{rp} + \sum_{q \in \Omega} \sum_{s=1}^{p-1} \sum_{k \in \Xi_{qs}} X_{qsk} \varepsilon_{kijg}) \\
& \quad + \sum_{r \in \Omega} \sum_{g \in \Gamma} \eta_{rg} (\hat{v}_g (A_{rg1} - 1) - R_{r1} + \sum_{j \in \Xi} v_{jg} X_{r1j}) \\
& + \sum_{r \in \Omega} \sum_{p \in \Pi'_r} \sum_{i \in \Xi_{r,p-1}} \sum_{j \in \Xi_{rp}} \sum_{g \in \Gamma} \lambda_{rpig} (d_{ijg} (A_{rpg} + X_{rpj} + X_{r,p-1,i} - 2) - D_{rp}) \quad (34)
\end{aligned}$$

subject to:

$$\begin{aligned}
& (22), (23), (28), (29), (30), (31), (32) \\
& R_{rp}, D_{rp} \geq 0 \quad \forall r \in \Omega, \forall p \in \Pi_r \quad (35)
\end{aligned}$$

Any feasible solution to problem VPD provides a lower bound to problem VPMIP. We therefore seek the tightest bound Z_D^* , the optimal solution to problem VPB, which is defined as follows:

$$(VPB) \quad Z_D^* = \max_{\mu \geq 0, \eta \geq 0, \lambda \geq 0} Z_{LG}^*(\mu, \eta, \lambda) \quad (36)$$

In order to assess the quality of the bounds that can be obtained from the Lagrangean relaxation, we denote:

$$\Upsilon = \{(X_{rpj}, A_{rpg}, D_{rp}, R_{rp}) \quad s.t. \quad (22), (23), (28), (29), (30), (31)\} \quad (37)$$

$$\Upsilon_{INT} = \{(X_{rpj}, A_{rpg}, D_{rp}, R_{rp}) \in \Upsilon \quad s.t. \quad (32)\} \quad (38)$$

$$A = \{(X_{rpj}, A_{rpg}, D_{rp}, R_{rp}) \quad s.t. \quad (24), (25), (26)\} \quad (39)$$

In general, it can be shown (see for instance Bertsimas and Tsitsiklis (1997)) that

$$Z_{LP}^* \leq Z_D^* \leq Z_{MIP}^*, \quad (40)$$

where Z_{LP}^* denotes the optimal value of the LP relaxation of problem VPMIP. Equality is obtained on the left of (40) if $CH(\Upsilon_{INT}) = \Upsilon$, where $CH(\cdot)$ denotes the convex hull of a set. This condition applies to the proposed formulation, because constraints (22), (23), (28), (29), (30), and (31) correspond to a totally unimodular matrix. The same argument

applies to the subproblems that are formed at each node in the branch and bound tree. Therefore, any advantage that is obtained from the Lagrangean relaxation will derive from the ability to obtain good primal solutions quickly.

Equality is obtained on the right of (40) if and only if $CH(\Upsilon_{INT} \cap A) = CH(\Upsilon_{INT}) \cap A$. For the formulation at hand, this last equality will generally not hold, and a duality gap will be unavoidable.

3.4 Solution methodology

We propose an algorithm to solve problem VPMIP. This algorithm consists of three tiers: a branch and bound framework that generates smaller subproblems of VPMIP by assigning values to some of the X_{rpj} and A_{rpg} variables; a subgradient optimization algorithm that obtains lower bounds to the corresponding subproblem; algorithms to solve an assignment problem, the shortest path problem, and a trivial decision problem. Each tier is now described in detail.

3.4.1 The branch and bound framework

Exploration of the branch and bound tree can be divided into two phases. The first phase assigns values to the X_{rpj} variables, and the second phase assigns values to the A_{rpg} variables. A sample branch and bound tree is presented in Figure 4. The heavy line joins the current node to the root node. All assignments that have been made at active nodes (nodes lying on the heavy line) are retained at the current node.

During the first phase of exploration, the X_{rpj} variables are ordered by their indexes r and p , so as to consider the reach stackers in round-robin fashion. Consequently, the first slot of all reach stackers is examined before proceeding to the second slot of any reach stacker, and so on. At each node, the algorithm generates an ordering of the containers that could be assigned to slot p of reach stacker r . This list only includes containers that have not been assigned to a slot by another active node. The available containers are sorted according to the minimum number of rehandles required to access them. At the root node, this number is equal to $\min_j v_{jg}$. As the algorithm progresses deeper into the tree, the minimum number of rehandles required to reach an unassigned container is dynamically

adjusted to reflect the impact of the assignments performed at other active nodes. This adjustment of the exploration order is critical to the performance of the algorithm.

A key observation is that it is never necessary to branch on the second portion of the tree. When the algorithm reaches the last node of the first phase, all containers have been assigned to a slot. All that remains to be established is the approach side when removing each container. The optimal value for the subtree to the right of this node is therefore obtained by solving a shortest path problem for each reach stacker. The nodes in this shortest path problem consist of two images of each container – one image for the water-side approach, and one image for the land-side approach. The single arc joining two nodes has a cost given by $h_{ijg} + r_{ijg}$, the distance and rehandling penalties incurred when the approach transition A_{rpg} is as implied by the two images in question. Each arc in the shortest path problem can be seen to correspond to a single A_{rpg} variable – if the arc is used in the shortest path, the corresponding A_{rpg} variable is set to one.

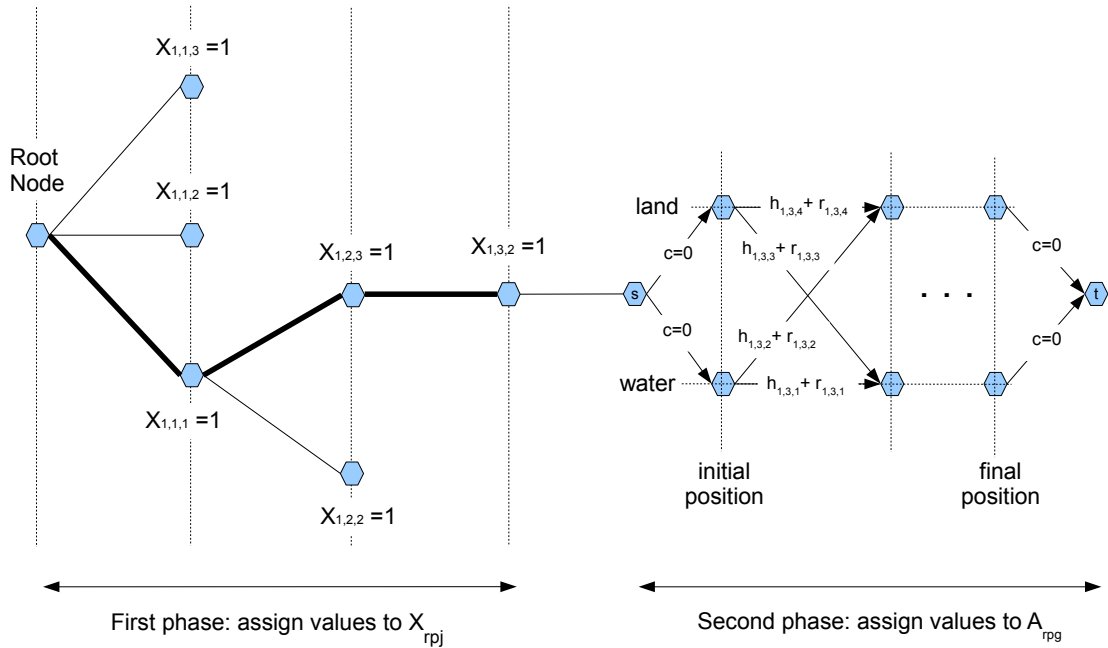


Figure 4: Sample branch and bound tree for a single reach stacker filling three slots

We obtain an important gain in computational efficiency by observing that many constraints become irrelevant at each node to the right of the root node. When an active node has set $X_{rpj} = 0$, we can eliminate any constraint that includes these variables within a *big M*-type term. All nodes to the right of the current node can use this information to reduce the number of constraints of types (24), (25), and (26) that need to be considered. Naturally, when the algorithm backtracks in the tree, the assignments made at each node are nullified, and the corresponding constraints must be reintroduced into the problem.

At each node in the tree, we obtain a subproblem $VPMIP^n$, which consists of finding optimal values to the subset of variables that have not been fixed, subject to the constraints that remain active. We obtain a lower bound for the optimal value of this subproblem by solving (perhaps approximately) the corresponding problem VPB^n . When this lower bound exceeds the objective value of a previously obtained feasible solution to $VPMIP$, we eliminate the current node, and all nodes arising from it, from further consideration.

3.4.1.1 Heuristic exploration of the branch and bound tree

Practical experience with the VPMIP indicates that a large number of optimal or near-optimal solutions may be found in the branch and bound tree. The bounds provided by the Lagrangean relaxation typically exhibit a duality gap of between 5% to 30% of the optimal value. Consequently, it is generally not possible to fathom any portion of the tree that contains a near-optimal solution. A heuristic backtracking mechanism is therefore proposed to avoid a time-consuming exploration of every portion of the branch and bound tree. The algorithm spends a user-specified amount of time seeking an improvement in the objective function. If no improvement is obtained within this time, the algorithm backtracks by a set number of nodes, effectively fathoming the branches that failed to yield an improvement. The number of nodes to backtrack grows with the amount of time spent since an improvement was last found, so that eventually the algorithm will backtrack to the root node and terminate. Clearly, this mechanism may prematurely dismiss a portion of the tree and fail to obtain the optimal solution. In Section 3.5, it will be shown that this problem is not a cause for great concern.

3.4.2 The subgradient optimization algorithm

Bertsekas (1999) describes the subgradient optimization algorithm and its use in conjunction with Lagrangean relaxation. At each iteration t , the algorithm computes a vector of Lagrangean multipliers $(\mu^t, \eta^t, \lambda^t) = \max\{0, (\mu^{t-1}, \eta^{t-1}, \lambda^{t-1} + \theta_t s^{t-1})\}$, where $\theta^t = \alpha^t(\hat{Z}_D - Z_D(\mu^t, \eta^t, \lambda^t))/|s^t|^2$; $0 < \alpha < 2$; s^t is the difference between the right-hand side and the left-hand side of the constraints computed at step t ; and \hat{Z}_D is an estimate of the optimal value of Z_D . The algorithm stops when the lower bound Z_D converges within a specified precision, or after a fixed number of iterations.

A significant computational advantage is obtained by noting that a large number of the dual values $(\mu^t, \eta^t, \lambda^t)$ generally take the value zero. When the dual value of constraint i is zero at an iteration, and $s_i^t \leq 0$, the dual value will retain a value of zero at the following iteration. The algorithm can then be limited to updating dual values that are currently positive, or where the corresponding difference $s_i^t > 0$. The latter condition can be readily detected, as it can only occur if the corresponding value of D_{rp} or R_{rp} is set to its lower bound. By monitoring the values of D_{rp} and R_{rp} , which can be done very efficiently, it is possible to predict which dual values will need to be updated.

3.4.3 Network algorithms

At each iteration in the subgradient optimization algorithm, we compute new values for $(\mu^t, \eta^t, \lambda^t)$, and solve a bipartite assignment problem, a shortest path problem, and a simple decision problem. The bipartite assignment problem is given by:

Minimize

$$\begin{aligned}
Z_1(\mu, \eta, \lambda) = & \alpha_w \sum_{r \in \Omega} \sum_{p \in \Pi_r} \sum_{j \in \Xi_{rp}} l_{rp} w_j X_{rpj} + \\
& \sum_{r \in \Omega} \sum_{p \in \Pi'_r} \sum_{i \in \Xi_{r,p-1}} \sum_{j \in \Xi_{rp}} \sum_{g \in \Gamma} \mu_{rpijg} \sum_{q \in \Omega} \sum_{s=1}^{p-1} \sum_{k \in \Xi_{qs}} X_{qsk} \varepsilon_{kijg} + b_{ijg} (X_{rpj} + X_{r,p-1,i}) + \\
& \sum_{r \in \Omega} \sum_{g \in \Gamma} \eta_{rg} \left(\sum_{j \in \Xi} v_{jg} X_{r1j} \right) + \\
& \sum_{r \in \Omega} \sum_{p \in \Pi'_r} \sum_{i \in \Xi_{r,p-1}} \sum_{j \in \Xi_{rp}} \sum_{g \in \Gamma} \lambda_{rpijg} d_{ijg} (X_{rpj} + X_{r,p-1,i}) \quad (41)
\end{aligned}$$

subject to:

$$(22), (23)$$

$$X_{r pj} \in \{0, 1\} \quad \forall r \in \Omega, \forall p \in \Pi_r, \forall j \in \Xi_{rp} \quad (42)$$

This problem is solved using a specialization of the successive shortest path algorithm which solves the assignment problem in polynomial time (Ahuja et al., 1993).

The shortest path problem arises from our need to define an approach direction for each reach stacker r at each step p in its loading sequence. The problem solved at each iteration of the subgradient method is given by:

Minimize

$$Z_2(\mu, \eta, \lambda) = \sum_{r \in \Omega} \sum_{p \in \Pi'_r} \sum_{i \in \Xi_{r,p-1}} \sum_{j \in \Xi_{rp}} \sum_{g \in \Gamma} (\mu_{rpijg} b_{ijg} A_{rpg}) + \sum_{r \in \Omega} \sum_{g \in \Gamma} (\eta_{rg} \hat{v}_g A_{rg1}) + \sum_{r \in \Omega} \sum_{p \in \Pi'_r} \sum_{i \in \Xi_{r,p-1}} \sum_{j \in \Xi_{rp}} \sum_{g \in \Gamma} (\lambda_{rpijg} d_{ijg} A_{rpg}) \quad (43)$$

$$(44)$$

subject to:

$$(22), (23), (28), (29), (30), (31)$$

$$A_{rpg} \in \{0, 1\} \quad \forall r \in \Omega, \forall p \in \Pi_r, \forall g \in \Gamma \quad (45)$$

This problem is equivalent to solving a shortest path problem on a suitably defined network. We can represent the water-side and land-side attack directions at each step as nodes in a network, and the variables A_{rpg} as arcs joining these nodes, as in Figure 5. For a reach stacker loading $|\Pi_r|$ containers, we will have $2|\Pi_r| + 2$ nodes, and $4|\Pi_r|$ arcs. We insert an artificial node s as the source, and an artificial node t as the sink, as shown in Figure 5. The source is connected by arcs of cost 0 to two nodes representing the starting position of the reach stacker. The sink node is connected to the nodes corresponding to the final position of the reach stacker, again using arcs with 0 cost. All other arcs have a cost given by the reduced cost ca_{rpg} of the corresponding variable A_{rpg} .

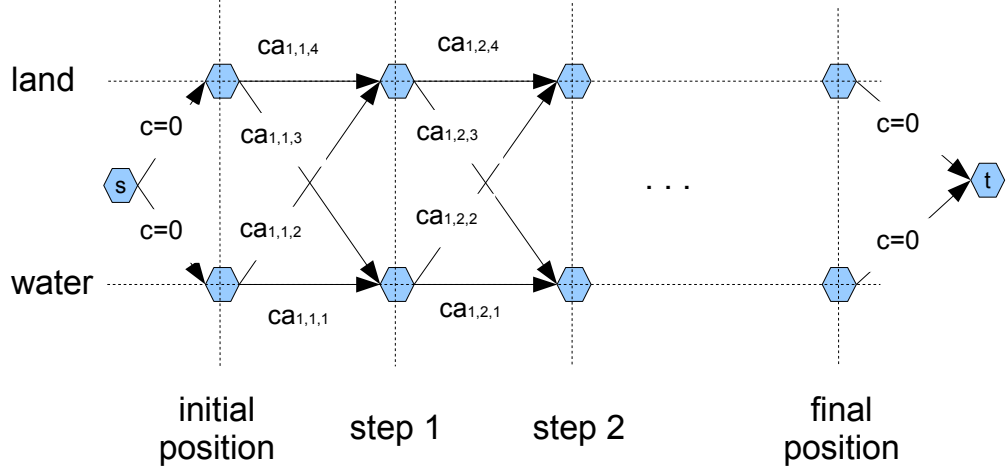


Figure 5: Shortest path problem in variables A_{rpg}

The algorithm of Dijkstra (1959) is used to solve the shortest path problem, given that the reduced costs of the variables A_{rpg} are non-negative. Note that the shortest path problem can be solved independently for each reach stacker.

The third, and simplest, problem is given by:

Minimize

$$\begin{aligned}
 Z_3(\mu, \eta, \lambda) = & \alpha_s \sum_{r \in \Omega} \sum_{p \in \Pi'_r} R_{rp} - \sum_{p \in \Pi_r} \sum_{i \in \Xi_{r,p-1}} \sum_{j \in \Xi_{rp}} \sum_{g \in \Gamma} \mu_{rpijg} R_{rp} - \sum_{g \in \Gamma} \eta_{rg} R_{r1} + \\
 & \alpha_d \sum_{r \in \Omega} \sum_{p \in \Pi'_r} D_{rp} - \sum_{p \in \Pi_r} \sum_{i \in \Xi_{r,p-1}} \sum_{j \in \Xi_{rp}} \sum_{g \in \Gamma} \lambda_{rpijg} D_{rp} \quad (46)
 \end{aligned}$$

subject to:

$$(35)$$

If the reduced cost of a variable R_{rp} or D_{rp} is positive, we set its value to 0. If the reduced cost is negative, we attempt to make it as large as possible. As stated above, the problem is unbounded, since we can set any variable with negative reduced cost to an

arbitrarily large number. Consequently, the following constraints are introduced at each node in the branch and bound tree, where the tilde notation signifies variables that have been set by any of the active branch and bound nodes. Note that these constraints are redundant for the corresponding subproblems $VPMIP^n$.

$$R_{rp} \leq \max_{i \in \Xi_{r,p-1}, j \in \Xi_{rp}, g \in \Gamma} b_{ijg} (\tilde{A}_{rpg} + \tilde{X}_{rpj} + \tilde{X}_{r,p-1,i} - 2) - \sum_{q \in \Omega} \sum_{s=1}^{p-1} \sum_{k \in \Xi_{qs}} \tilde{X}_{qsk} \varepsilon_{kijg} \quad \forall r \in \Omega, \forall p \in \Pi'_r \quad (47)$$

$$R_{r1} \leq \max_g \sum_{j \in \Xi_{r1}} v_{jg} \tilde{X}_{r1j} - \hat{v}_g (1 - \tilde{A}_{r1g}) \quad \forall r \in \Omega \quad (48)$$

$$D_{rp} \leq \max_{i \in \Xi_{r,p-1}, j \in \Xi_{rp}, g \in \Gamma} d_{ijg} (\tilde{A}_{rpg} + \tilde{X}_{rpj} + \tilde{X}_{r,p-1,i} - 2) \quad \forall r \in \Omega, \forall p \in \Pi'_r \quad (49)$$

Although the variables X_{rpj} were decoupled from the variables A_{rpg} , and their values obtained in separate subproblems, they can be merged back into a solution to $VPMIP^n$ without conflict. On the other hand, the variables D_{rp} and R_{rp} may not be valid for problem $VPMIP^n$ given the values that were separately obtained for X_{rpj} and A_{rpg} , because setting any of the variables R_{rp} or D_{rp} to zero may result in an infeasible solution overall. A feasible solution is readily obtained by inspecting all active constraints (24), (25), (26), and increasing the values of R_{rp} and D_{rp} as necessary.

3.5 Computational results

We present three sets of experiments in this section. The first set of experiments provides a performance benchmark against CPLEX 9.0 (ILOG S.A., 2003). The second set of experiments is designed to estimate the error that is incurred by applying the heuristic described in Subsection 3.4.1.1. The third set of experiments evaluates the performance of the proposed algorithm using actual vessel plans. The proposed algorithm was implemented in the C++ programming language. All experiments were performed on a 1.6 MHz Pentium computer with 512 MB RAM.

For the first set of experiments, problem VPMIP was formulated using AMPL (Fourer

et al., 2003). The node-arc notation available in AMPL was used to formulate constraints (22), (23), (28), (29), (30), and (31). This notation signals to CPLEX the presence of an embedded network structure, triggering the use of the more efficient network simplex method on portions of the problem.

The data used for the first set of experiments were limited to ten or fifteen containers (it was not practical to solve larger instances using CPLEX 9.0). The common data for all experiments specify three classes of containers, with the containers distributed on three stacks in the yard. The stowage plan called for all containers to be placed in a single hold of the ship. Each replication of the experiments called for different stowage positions for each of the three container classes.

penalty coefficients	LP relax. time	Root Node Examination			DFS time to opt
		Best primal	Time best primal	Bound	
$\alpha_d \approx \alpha_s \approx \alpha_w$	1.2 s	103%	0.40 s	23%	0.87 s
$\alpha_d > \alpha_s > \alpha_w$	1.4 s	103%	0.36 s	21%	0.78 s
$\alpha_d > \alpha_w > \alpha_s$	1.9 s	105%	0.32 s	26%	0.70 s
$\alpha_s > \alpha_d > \alpha_w$	1.1 s	122%	0.44 s	34%	0.75 s
$\alpha_s > \alpha_w > \alpha_d$	0.9 s	103%	0.48 s	45%	0.79 s
$\alpha_w > \alpha_d > \alpha_s$	2.1 s	103%	0.33 s	32%	0.81 s
$\alpha_w > \alpha_s > \alpha_d$	0.8 s	102%	0.43 s	34%	0.67 s

Table 2: Performance benchmark using ten containers

Different values of α_s , α_d , and α_w were used in the experiments to reflect the different planning preferences that may be encountered in practice. The replications were further grouped according to the values of α_s , α_d , and α_w employed. For the scenario with ten containers, a total of 210 replications were performed, divided into seven such subgroups. For the scenario with fifteen containers, a total of 21 replications were performed.

The results of these experiments are presented in Tables 2 and 3. The first column in these tables indicates the relative importance of the rehandling, ground distance, and weight distribution penalties. The second column indicates the time required by CPLEX 9.0 to obtain an optimal solution. The three columns that follow refer to values obtained at the root node of the branch and bound tree. These columns indicate the best primal

solution obtained (as a percentage of the true optimal value), the time required to obtain this solution, and the best lower bound obtained for the problem (again, as a percentage of the true optimal value). The last column indicates the time required to obtain the optimal solution using depth-first search following exploration of the root node. The heuristic mechanism described in Subsection 3.4.1.1 was disabled for these experiments.

penalty coefficients	LP relax. time	Root Node Examination			DFS time to opt
		Best primal	Time best primal	Bound	
$\alpha_d \approx \alpha_s \approx \alpha_w$	289 s	113%	1.4 s	13%	6.6 s
$\alpha_d > \alpha_s > \alpha_w$	555 s	104%	0.3 s	3%	4.2 s
$\alpha_d > \alpha_w > \alpha_s$	2236 s	105%	1.0 s	4%	13.2 s
$\alpha_s > \alpha_d > \alpha_w$	296 s	101%	0.5 s	5%	1.9 s
$\alpha_s > \alpha_w > \alpha_d$	419 s	102%	0.8 s	9%	1.8 s
$\alpha_w > \alpha_d > \alpha_s$	1393 s	106%	0.7 s	17%	4.1 s
$\alpha_w > \alpha_s > \alpha_d$	185 s	102%	0.6 s	37%	1.4 s

Table 3: Performance benchmark using fifteen containers

The results indicate that excellent primal solutions are quickly obtained at the root node. However, it is clear that the lower bounds obtained at the root node are of very poor quality. As the algorithm moves deeper into the branch and bound tree, however, the quality of the bounds increases dramatically. For these small experiments, the depth-first search algorithm performs most of its work at around 25% of the depth of the branch and bound tree – the bounds obtained at that level are sufficiently accurate to permit pruning of deeper nodes. For larger problems, the fact that only a fixed percentage of the tree is explored is of little comfort, as the size of the tree grows exponentially. This motivates the use of the heuristic proposed in Subsection 3.4.1.1.

The second group of experiments was performed using a data set with twenty containers. An exhaustive exploration of the branch and bound tree is still possible for problems of such size. Each replication consists of two parts. The first part obtains a near-optimal solution using the heuristic described in Subsection 3.4.1.1, with a time limit of two seconds before backtracking, and an overall limit of ten minutes. The second part of the replication performs a thorough search of the branch and bound tree without heuristic backtracking in

order to determine the true optimal solution.

Twenty replications were performed. In 18 instances, the heuristic method found the optimal solution. In the remaining two instances, the heuristic found, respectively, a solution within 1% and 15% of the optimal solution. The heuristic took an average of 4.5 minutes to find the best solution. Each exact exploration of the branch and bound tree took between two and six hours per replication.

The third set of experiments uses a data set with 172 containers. Obtaining the optimal solution for instances of this size is not practical. Furthermore, the lower bounds obtained at the root node do not provide a reliable estimate of the optimality gap. An attempt is made to estimate the optimality gap in larger instances by solving two simpler problems. The first problem minimizes only the weight distribution component of the objective function, ignoring any constraints relating to ground travel or rehandling. This amounts to solving a bipartite matching problem, which can be done very quickly. The second problem minimizes ground travel with no regard for rehandling or weight distribution penalties. This problem is solved without integer variable restrictions. The solutions obtained for these two problems provide a rough indication of the quality of the solution obtained for problem VPMIP.

Eighteen replications were run, each for a total of twenty minutes. As shown in Table 4, the results are within 5% of the solution for the single-objective relaxation focusing on weight distribution, and within 16% of the single-objective relaxation focusing on ground travel. These replications focus on situations where rehandles are given greater importance than ground travel and vessel stability considerations. When the planner’s priorities emphasize vessel stability or ground travel, acceptable solutions can be obtained using a simple heuristic.

coefficients	time	Gap to single-obj weight	Gap to single-obj travel
$\alpha_s > \alpha_d \approx \alpha_w$	10.9 min	105%	116%
$\alpha_s > \alpha_d > \alpha_w$	9.4 min	105%	114%
$\alpha_s > \alpha_w > \alpha_d$	19.7 min	105%	115%

Table 4: Estimation of the optimality gap for problems with 172 containers

3.6 Conclusions

This chapter presented an application of mathematical programming techniques to the automation of the vessel planning process in maritime terminals using reach stackers. The models and techniques that have been described can be adapted to other types of stacking equipment, such as RTGs and RMGs.

The performance of the proposed algorithm depends chiefly on three elements: the speed with which each iteration of the subgradient algorithm is completed, the quality of primal solutions uncovered at each node, and the quality of the bounds obtained at each node. The proposed algorithm exploits the special structure of the problem to great advantage, greatly limiting the number of constraints that are explored, and using efficient network algorithms to solve each subproblem. The algorithm can also obtain excellent primal solutions. In small problems, the algorithm obtains results that are very close to the optimal value simply by inspecting the root node. In problems of practical size, however, iterations at the root node can be quite slow due to the large number of constraints that must be examined. It is therefore advantageous to move along the branch and bound tree, where thousands of subproblems can be solved in a few seconds.

The quality of the bounds obtained near the root node is quite poor, but improves considerably as the algorithm progresses into the second quarter of the branch and bound tree. Nevertheless, the bounds obtained by the proposed algorithm are not strong enough to permit a sufficiently effective fathoming of the tree. A heuristic ejection mechanism was proposed to limit the amount of time spent in a given portion of the tree. This mechanism was seen to accelerate the algorithm, generally at very little cost to the quality of the final solution.

Given the uncertainty associated to the arrival times of export containers, the vessel planning algorithm is best applied over limited time horizons, as in the Δ -replan technique proposed by Steenken et al. (2001). The proposed algorithm is well suited for such an application.

Chapter 4

A Heuristic Approach to the Vessel Planning Problem

The vessel planning problem was described in detail in Chapter 3. In this chapter, we propose a heuristic algorithm for the vessel planning problem. The heuristic methods described in the OR literature are particularly well suited to the problems that arise in maritime logistics and operations. This follows from the highly combinatorial nature of these problems, and the difficulty of expressing these problems in traditional mathematical modeling language. Heuristic methods in OR differ from exact mathematical formulations in that the former are not guaranteed to reach an optimal solution, although they generally produce remarkably good solutions very quickly (Glover and Kochenberger, 2003).

The central contribution of this chapter is a metaheuristic for automated load planning in container terminals with deep blocks. Due to the complexity and variability of the maritime terminal environment, the participation of the ship planner is essential to ensuring a safe loading operation. Consequently, this chapter also proposes a decision support tool to accompany the core metaheuristic algorithm. While the core algorithm automates most of the processes, the decision support tool allows an experienced planner to refine or overrule the automated vessel plan as needed.

The remainder of this chapter comprises four sections. Section 4.1 presents a heuristic algorithm to solve the vessel planning problem. Section 4.2 presents a graphical user interface

and decision support tool to accompany the core algorithm. Section 4.3 presents results for computational experiments based on actual data from a Spanish container terminal. Finally, Section 4.4 presents conclusions and directions for further work.

4.1 A heuristic for efficient vessel planning

The proposed heuristic algorithm is based on the tabu search methodology, a technique that has become popular due to its success in quickly obtaining good solutions to complex optimization problems (Glover and Laguna, 1997). Tabu search starts by generating an initial solution, which although feasible, is readily improved upon. Then, the algorithm iteratively explores alternatives to the current solution by generating *neighbors*. These are similar to the current solution, except for generally small modifications. The algorithm selects the best neighbor, which then becomes the current solution, and begins to explore the neighborhood of this new solution. Cycling is prevented by disallowing the algorithm from undoing its most recent actions, which are considered “tabu”. The algorithm keeps a fixed-length list of the most recent actions, the tabu list. After a fixed number of iterations are performed, or when no valid neighbors can be found, the algorithm returns the best plan visited.

This section describes the proposed algorithm using pseudo-code constructs (Cormen, Leiserson, and Rivest, 1990). The following notation is used in what follows:

$\eta = \Xi $	total number of containers to be loaded; total number of slots to be filled in the ship.
$N = \Omega $	number of available stevedoring crews.
$first_i$	number of first slot to be loaded by crew i .
$last_i$	number of last slot to be loaded by crew i .
m	number of initial solutions produced by multistart.
s_j	sequence in which container j will load.
q	number of neighbors to retain from the initial, larger list.

R	total number of boxes rehandled.
D	total ground distance traveled by reach stackers.
W	total instability measure.

High level pseudo-code for algorithm *shipPlanOptimization()* is presented below. First, the algorithm inputs all required data from databases or from the user interface. These data include: the spatial layout of the container yard; a list of containers currently in the yard, including container ID, type, length, and destination; geometry of the ship and its bays; general stowage plan, specifying the class of each slot to be loaded on the ship; and number of stevedoring crews available to load the vessel.

The algorithm checks the feasibility of the problem by ensuring that there are enough containers in the yard for each of the different types required in the stowage instructions. Throughout this discussion, it is assumed that, for any container class mentioned in the stowage plan, the number of containers required is exactly equal to the number of containers available in the yard.

Algorithm 1 *shipPlanOptimization()*

```

yard  $\leftarrow$  DB // includes yard geometry and containers
ship  $\leftarrow$  DB
N  $\leftarrow$  UI
stowagePlan  $\leftarrow$  DB
if verifyFeasibility(yard, ship, stowagePlan, N) then
  firstSolutions  $\leftarrow$   $\emptyset$ 
  bestSolutions  $\leftarrow$   $\emptyset$ 
  allocHoldsToCranes(N)
  autoSequenceSlots(firstSolutions, m)
  prioritizeYard(yard)
  best.cost()  $\leftarrow$   $\infty$ 
  for  $i = 1$  to  $m$  do
    firstSolutions[ $i$ ]  $\leftarrow$  createInitial( $i$ , firstSolutions[ $i$ ])
    bestSolutions[ $i$ ]  $\leftarrow$  tabuImprove(firstSolutions[ $i$ ])
    if bestSolution[ $i$ ].cost() < best.cost() then
      best  $\leftarrow$  bestSolutions[ $i$ ]
  return best

```

The proposed algorithm uses a technique known as multistart, whereby a variety of initial plans are generated and explored. This permits a more thorough investigation of the solution space. The main algorithm calls three functions to allocate the holds of the

ship amongst the available crews, and prepare the data structures required for the efficient generation of multiple initial solutions. The algorithm then calls a tabu search procedure to improve the quality of each one of the initial solutions generated. The best solution from the improved plans is selected and presented to the user. The remainder of this section describes the components of the algorithm in greater detail.

Procedure *allocHoldsToCranes()* partitions the ship’s holds amongst the available stevedoring crews. The goal is to distribute the work as evenly as possible amongst the crews. In its simplest form, the algorithm attempts to assign a similar number of containers to each crew, with the restrictions that only one crew can work on a given hold, and the holds assigned to a given crew must be contiguous. Other allocation mechanisms are readily implemented. For instance, the number of containers can be apportioned to each crew based on the number of hours the crew will be working on the ship. For each crew i , procedure *allocHoldsToCranes()* assigns values to $first_i$ and $last_i$.

The procedure *autoSequenceSlots()* assigns a number s_j from $first_i$ to $last_i$ to each slot j that will be loaded by crew i . The number s_j indicates the order in which the slot will be filled by the corresponding crew. Since each position in the sequence can be associated with a slot in the ship, at this point the algorithm has generated one ordered list of generic containers for each crew, as illustrated in Figure 6.

Bay:01 col: 04 row82 HKGHKG HR20 FULL SEQ: 1 CREW: 1	Bay:01 col: 02 row82 HKGHKG DC20 FULL SEQ: 2 CREW: 1	Bay:01 col: 07 row88 MALTPP HC20 FULL IMO:5.1 SEQ: 125 CREW: 1	Bay:17 col: 10 row12 ITLSPE HR20 FULL SEQ: 126 CREW: 2	Bay:17 col: 08 row12 ITLSPE DC20 FULL SEQ: 127 CREW: 2	Bay:17 col: 07 row84 ITLSPE HR20 FULL SEQ: 197 CREW: 2
---	---	---	---	---	---

Figure 6: Sequence of generic containers for two crews

The procedure *prioritizeYard()* assigns priority numbers to all the containers that are to be loaded, according to their placement in the yard. The goal is to produce sensible groups of containers on which the crews can focus without interfering with each other. Three different sorting operators are used in this regard: North-South, East-West, and attack side (land-side or sea-side). Ties are broken by giving precedence to containers that are more accessible – that is, containers closer to the aisles and higher up in the stacks. By

combining these three sort operators and alternating their direction, one can obtain nine¹ initial sortings of the containers. Each crew will prioritize the containers according to a different initial ordering.

Algorithm 2 createInitial (sortMethod, currSol)

```

for crew = 1 to  $N$  do
  for  $j = first_{crew}$  to  $last_{crew}$  do
    class  $\leftarrow$  currSol.slot( $j$ ).getClass()
    currSol.container( $j$ )  $\leftarrow$  yard.getContainer(class, crew + sortMethod)
  currSol.cost()  $\leftarrow$  computeFullCost(currSol)

```

Algorithm *createInitial()* constitutes the multistart component of the proposed methodology. The procedure generates multiple initial solutions by assigning different initial sortings to each crew. In this manner, the m generic arrays, which are initially identical, are matched to the containers in the yard in different ways. This results in m distinct initial load plans.

The procedure assumes that the stevedoring teams work simultaneously, and containers are removed from the yard according to their sequence number within each crew’s lot². This permits a realistic simulation of the clearing of the stacks. Based on the yard sorting methods described above and the current position of their reach-stacker, each crew uses the most immediately accessible container of the required class. Initial plans thus typically contain sequences of adjacent containers being loaded into adjacent positions in the ship. This method turns out to generate initial plans that are of good quality.

Algorithm *createInitial()* then computes the cost of the initial plans. The cost of a plan is computed following Equation 50, as the weighted sum of four components: travel of the reach-stacker on the ground, number of containers rehandled in the stacks, disutility due to interference between multiple crews, and a measure of the vertical weight distribution on the ship. The travel distances in the yard can be easily computed given the simple geometry of most yards. The number of rehandles is computed by simulating the work of the reach-stackers on the blocks. The algorithm keeps track of the containers that have

¹Currently, no more than eight quay cranes are assigned to a ship at the busiest terminals in the world: PSA Singapore and Hong Kong International.

²That is, at sequence step p , crew i will load into slot $first_i + p - 1$.

been loaded on the ship at any given point, thus providing an accurate count of the number of removal and restacking operations needed to retrieve any given container. Interference between crews is computed whenever two reach-stackers are scheduled to work within a certain distance of each other. Finally, the algorithm measures the disutility of the vertical weight distribution on the ship by multiplying the weight of each container by its distance from the ship’s bottom. This encourages plans where the lighter containers are placed in the higher slots.

$$z_{tabu} = \alpha_s R + \alpha_d D + \alpha_w W \tag{50}$$

After the initial solutions have been generated, the top-level algorithm calls procedure *tabuImprove()* to perform a local search starting from each initial solution. Each iteration of the search begins with the generation of a list of “neighbors” of the current plan. These neighbors are generated from the current plan by exchanging two sequences of equivalent containers, as shown in Figure 7.

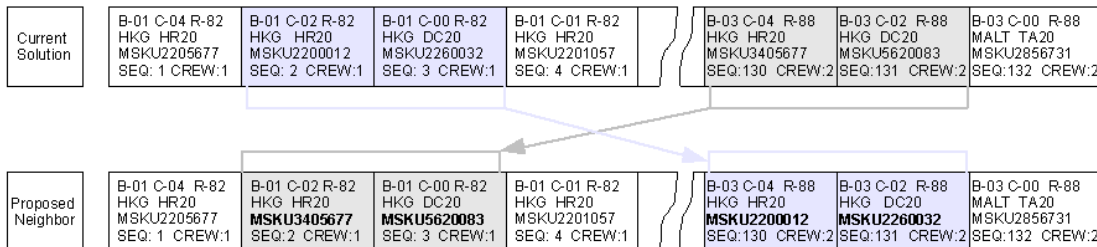


Figure 7: Modifying a plan to generate neighbors

Noting the multiple ways in which these permutations could be generated, it becomes clear that the number of neighbors could be extremely large. Furthermore, evaluating the cost improvement of each neighbor is computationally expensive, because one must recalculate the new state of the stacks and its impact on rehandles.

In the interest of computational efficiency, the neighborhood is generated in two steps. The first step, procedure *generateNeighbors()*, creates a large number of neighbors, but computes only a rough (and computationally trivial) approximation of the change in cost for each neighbor. The neighbors are ranked according to this cost proxy, and only the

q neighbors at the top of the ranking are retained. A more thorough calculation of the cost benefits of the q chosen neighbors is performed in procedure *getBestValid()*. The best candidate that is not listed in the tabu list is selected.

Algorithm 3 *tabuImprove* (*currSol*)

```

for  $i = 1$  to  $m$  do
   $tabuList \leftarrow \emptyset$ 
   $nbMoves \leftarrow 0$ 
   $validMove \leftarrow true$ 
  while  $validMove$  and ( $nbMoves < maxMoves$ ) do
     $neighborList \leftarrow generateNeighbors(currSol)$ 
     $neighborList \leftarrow trim(neighborList, q)$ 
     $bestNeighbor \leftarrow getBestValid(neighborList, tabuList)$ 
    if  $bestNeighbor \neq \emptyset$  then
       $currSol \leftarrow bestNeighbor$ 
       $tabuList \leftarrow tabuList.append(bestNeighbor)$ 
      if  $currSol.cost() \leq bestSol.cost()$  then
         $bestSol \leftarrow currSol$ 
    else
       $validMove \leftarrow false$ 
  return  $bestSol$ 

```

It is important to note that the quality of the solutions does not increase monotonically. The search can move out of local optima, allowing a broader exploration of solution alternatives. In any event, the “best-yet” solution is always stored. If the search wanders into less desirable alternatives, the algorithm can always revert to the best-yet plan. The algorithm terminates after a preset number of iterations, or when no new admissible neighbors can be found (this can happen, for instance, if all potential neighbors are represented in the tabu list). The algorithm outputs the best plan visited.

In forming the first list of neighbors, the algorithm scans the current solution for clusters of containers. To qualify for a cluster, a group of containers must be of the same class, be currently assigned to a single crew, and be located in the same module in the yard. In this way, the algorithm avoids breaking up “least cost” groups of containers that can be removed from the stack without any unnecessary rehandling, and without any displacement by the reach-stacker. Clearly, these segments can’t be improved upon, and any disruption of their continuity would increase the overall cost of the plan. The algorithm creates a list of clusters of the same size, which are candidates for swapping. Clusters can be swapped

Algorithm 4 generateNeighbors(sol)

```
prevClust  $\leftarrow$  0
currClust  $\leftarrow$  0
currYardModule  $\leftarrow$  0
length  $\leftarrow$  0
for  $i = 1$  to  $\eta$  do
  currClust  $\leftarrow$  concat(sol[i].class(), sol[i].crewNb(), sol[i].yardModule)
  if currClust  $\neq$  prevClust then
    blocksToExplore[length][currClass].push( $i - \text{length} - 1$ )
    length  $\leftarrow$  0
    prevClust  $\leftarrow$  currClust
  else
    length  $\leftarrow$  length + 1
for  $l \in$  blocksToExplore.keys1() do
  for  $c \in$  blocksToExplore.keys2() do
    pairs[l]  $\leftarrow$  blocksToExplore[l][c]  $\cdot$  blocksToExplore[l][c]
    pairs[l].sortBy(proxyCost)
```

between different crews.

4.2 A decision support system for vessel planning

Whenever modeling a real-life problem using mathematical and computational methods, several abstractions are necessary to make the problem tractable. While remaining quite close to the problem at hand, this chapter has presented a somewhat simplified version of the problem actually faced by the ship planners. Any number of complications and variations can arise in practice. For instance, export containers frequently arrive at the yard at the very last moment; stowage plans may need revision due to violations of IMO recommendations or structural constraints on the ship; stevedoring crews may be delayed or altogether unavailable; etc.

The problem described in this chapter is multi-objective, in that the quality of the plan is judged according to four measures of disutility. The algorithm proposed here is not truly multi-objective, as it combines the four measures of disutility into a single cost. The trade-offs implicit in this weighting method are not always the best reflection of the stakeholders' true preferences. Closer consideration of the alternative plans by an experienced planner thus becomes necessary.

Experienced terminal staff often use clever shortcuts to expedite work in the yard. These shortcuts are inconsistent with the procedural assumptions made here. For instance, the

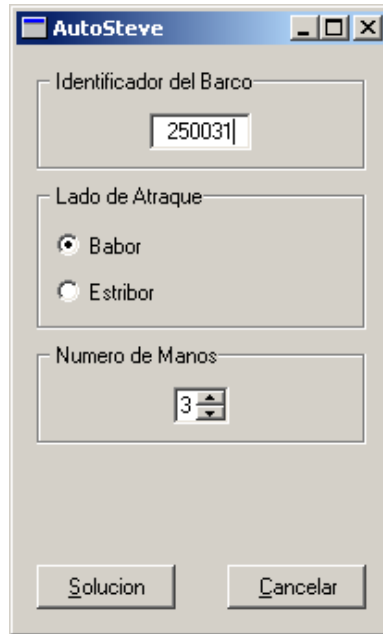


Figure 8: Initial screen of the DSS

planner may instruct the crane operator to temporarily depart from the established layering pattern for containers on board. The planner may also ask the reach-stacker operators to temporarily store containers in unmarked areas. Reach-stacker operators sometimes access containers from the side, rather than the front of a stack. Representing these “tricks of the trade” formally within the algorithm could be quite difficult, and would surely be detrimental to the computational performance of the algorithm.

The proposed metaheuristic does not address the issues discussed in the three preceding paragraphs. Rather, the algorithm specializes in the solution of a well-defined and slightly simplified problem. The decision support system (DSS) that calls the metaheuristic, on the other hand, allows the planner to make last-minute changes to the plan proposed by the algorithm, while receiving prompt feedback from the tool regarding the feasibility and estimated cost of those modifications.

When the DSS launches, it obtains some basic information from the user: the ship ID, the number of crews that are available, and the position of the ship on the quay (starboard or port). A screenshot of the initial DSS user interface is presented in Figure 8. The DSS then establishes connections to the relevant databases and retrieves information on the current

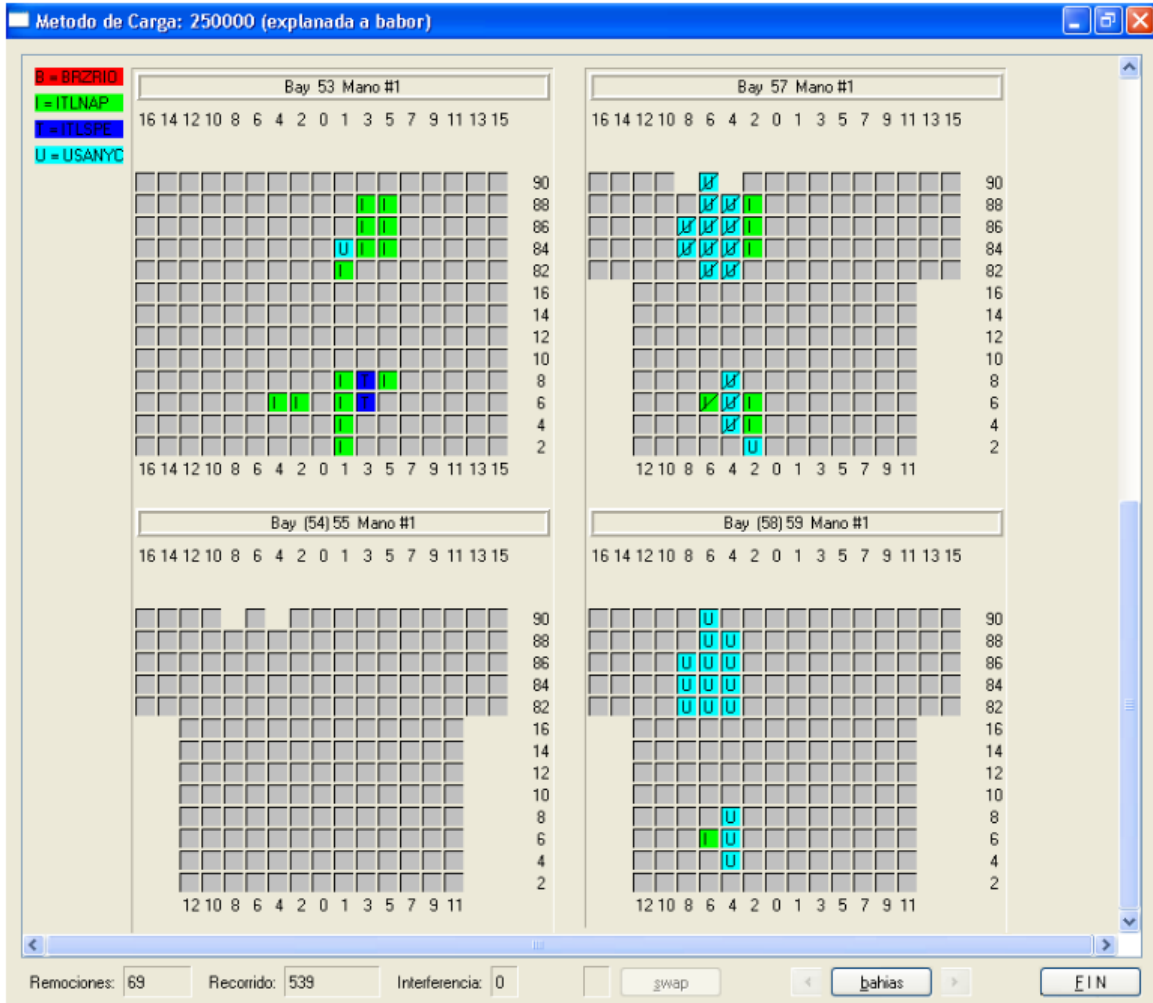


Figure 9: Full-ship solution screen of the DSS

state of the yard, the ship’s geometry, and the stowage plan. The core metaheuristic is called, and the best solution found is returned. The DSS presents the plan in a graphical format that is familiar to the planners, as shown in Figure 9. The planner can observe the plan at the level of the entire ship, or by ship hold. The DSS provides icon-driven commands to change the loading position of a container. When the planner attempts to change the proposed solution to one that is deemed to have a higher cost, a warning message is shown, giving the planner the alternatives of continuing, or cancelling the action. When the planner is satisfied with the load plan, it is saved to a database. The final plan must be approved by the ship’s captain, who is ultimately responsible for the stability of the vessel. The stevedoring foremen use printed and electronic versions of the finalized plan

while loading the vessel.

4.3 Experimental results and analysis

The algorithm was tested throughout the implementation process using actual data from ten ships loading a small number of containers (30-100 TEUs) at a Spanish port. This guided the development effort by shedding light on the benefits and computational requirements of different variants of the algorithm. Several experienced ship planners reviewed the results and offered their guidance in this respect.

Having finalized the fundamental design of the algorithm, a second set of experimental runs was performed. This set of tests fulfilled three objectives. First, it permitted a formal assessment of the benefits of the multistart technique. Second, it allowed for an understanding of the trajectory of the objective function as the algorithm progresses. Finally, the tests illustrated the impact of the neighborhood size on the solution quality.

The data for these tests were based on three ships of 2,000, 4,500, and 7,000 TEU, and loading 74, 407, and 591 containers, respectively. Additional cases were generated by bootstrapping the three base cases (by exchanging the location of no more than 10% of the ship's containers). The tests were run using five multistart plans. These tests were done on an 800 MHz Pentium III computer.

All runs were completed in one to five minutes. The results indicate that the initial plans may be quite dissimilar, with objective values differing by up to 15%. The search algorithm can improve an initial plan by up to 35%, but occasionally the improvement is as low as 1%. The best initial plan does not always lead to the best final plan, and the search process can transform similar initial plans into very dissimilar final plans. These observations indicate that multistart is indeed valuable. For larger ships, the quality of the solution would likely benefit from a larger number of initial plans.

As is often observed with tabu search, in these experiments the objective function improves rapidly during the initial iterations, and then oscillates within a relatively narrow range. After a small number of iterations (about 15 for the small ship and 40 for the larger ships), the algorithm has achieved most of the improvements it would likely ever achieve (as

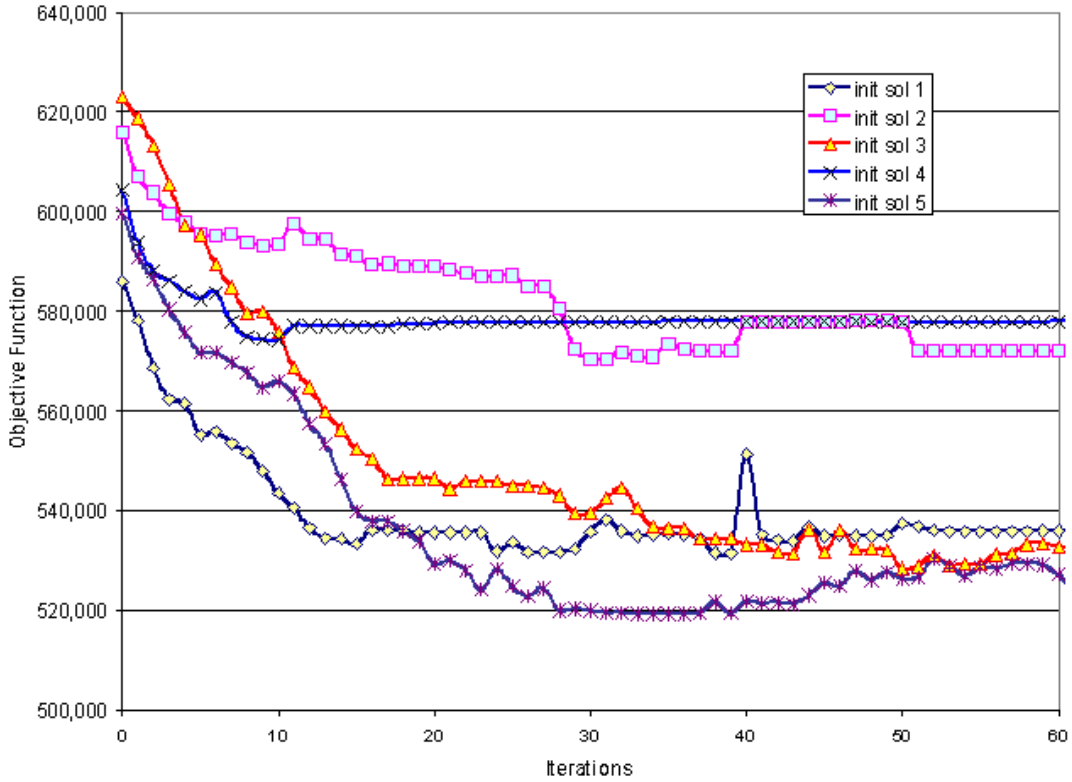


Figure 10: Objective function trajectory for 4,500 TEU ship

verified by running the algorithm for a very large number of iterations). This is illustrated in Figure 10. It is thus plausible to run the algorithm until either convergence is achieved, or for a problem-specific number of iterations. The speed of convergence will depend on the size of the neighborhoods. For this problem, generating large neighborhoods is computationally expensive, and it is therefore preferable to perform a larger number of iterations on smaller neighborhoods.

As mentioned in Section 4.1, the algorithm chooses the best neighbor using a two-phase selection process. Since the cost proxy used in the first phase is an approximation, many low-quality neighbors are initially ranked incorrectly as cost-improving options. If too many neighbors are generated, these “false positives” displace the truly cost-improving neighbors from the ranking, and choke the selection process. It is therefore important to provide sufficient space in the ranking vector to allow for this effect. Other techniques, such as stochastic selection of neighbors are promising in this regard.

In a third set of experiments, the MIP formulation presented in Section 2.2 was implemented using AMPL 9 (Fourer et al., 2003), and solved using CPLEX 9.0 (ILOG S.A., 2003). These experiments were run on a 1600 MHz Pentium M computer with 512 MB of RAM. An initial set of tests, with 30, 50, and 100 containers was prepared. The results resonate with the statements of Bish et al. (2005), who found that a MIP formulation of their problem takes several hours to solve problems with as little as twenty containers. For the experiments described here, no instance with more than 30 containers was successfully *presolved*³ by AMPL. The memory requirements of the presolve phase exceeded the test machine’s RAM, forcing computations to page to the hard drive. The consequent impact on performance forced the interruption of these runs after several hours in the presolve phase.

A small example with twenty containers was successfully presolved by AMPL and passed to CPLEX. This problem was solved by the metaheuristic in less than one second. The value of the objective function reported by the metaheuristic was compared to the values reported by CPLEX after 2, 20, and 200 seconds in the solve phase. After 2 seconds, the MIP’s objective function was twice as high as that reported by the metaheuristic. After 20 seconds, the MIP outperformed the metaheuristic by 5%. At 200 seconds, the MIP outperformed the metaheuristic by 40%.

In the latter case, the disadvantage of the metaheuristic appears to come from its inability to understand the extended impact of changing attack sides. When forming neighborhoods, the algorithm focuses on creating clusters of containers and sequencing those clusters. The decision of attacking each block land- or sea-side is subordinate to sequencing. A comparison against the plan generated using the MIP would suggest increasing the visibility of the attack-side decision process.

4.4 Conclusions and directions for further research

This chapter presented an application of tabu search and multistart to the automation of the vessel load planning process in maritime terminals. The proposed algorithm is specifically

³During the presolve phase, AMPL attempts to simplify the original model by removing redundant constraints and replacing variables that are constrained to a single value.

tailored to terminals that operate their yards with reach stackers. The core algorithm was embedded in a decision support system to permit a higher degree of control on the part of the terminal staff. The resulting DSS has been informally praised by experienced ship planners based on the results obtained for a set of real-world tests. The algorithm finds good solutions very quickly, and can solve problems with hundreds of containers in a few minutes.

Based on the experimental runs, it appears that the neighbor generation mechanism could be improved to make better decisions regarding the attack direction at each block. Currently, the algorithm is too myopic in this regard, and consequently fails to generate solutions with adequate attack-side transitions. A bi-level approach similar to that proposed by Kim et al. (2004) is attractive in this sense: the high-level component would choose an area of the block and an attack direction, while the low-level component sequences the containers in that sector.

There exist numerous directions for further research. First, further comparison of the plans produced by the algorithm against those produced manually is desirable. A more tractable definition of what constitutes acceptable tradeoffs between the multiple measures being minimized in the planning process must be sought and coded into the algorithm. In this regard, it may be possible to extend the algorithm to a true multi-objective search (Jones et al., 2002). This approach would allow the decision makers to better understand the qualities of the different planning options, and guide the algorithm towards the preferred solution space. At this point, however, this is not a problem in algorithm design – there is simply no consensus in the maritime world about the relative value of the different objectives.

The findings in this research indicate that standard MIP formulations are inadequate for solving actual instances of the vessel load planning problem. The basic MIP formulation is unable to address problems with as little as 35 containers. In order to perform a plausible calibration of the metaheuristic against established MIP solution software, more sophisticated modeling techniques will be necessary. In this regard, branch-and-bound and

cutting plane strategies may prove useful. The combination of traditional mathematical programming methods with metaheuristics is also a promising direction of research (Carrotta, 2005).

Currently, the algorithm assumes that multiple crews load containers at the same rate. Although this is likely to be a good first-degree approximation, this assumption should be tested with a simulation model. This model would mimic the loading plan indicated by the metaheuristic, but use actual travel times and introduce random variations in the quay cranes' load cycles. The validity of the original assumption can be understood by estimating the additional congestion in these cases.

The speed of the algorithm is satisfactory, largely due to the use of cost proxies in neighborhood generation. Further enhancements to the accuracy of the cost proxy are possible. Regression testing of different proxy formulae would be a straightforward manner of reaching this goal.

The user interface component can be enhanced in a great variety of ways, without significant changes to the core algorithm. For instance, the interface may be extended to better accommodate export containers that arrive late to the yard. Finer control of the crane split decisions would likely be of great value to terminal planners. Finally, more powerful methods for moving containers to new positions would enhance the user experience.

Some of the ideas presented in this chapter are applicable to terminals operating with different types of stacking equipment. Most terminals in the Far East operate mixed fleets of stacking equipment, with an emphasis on rubber-tired gantry cranes (Alvarez, 2006). There is currently a manifest demand for algorithms to optimize the utilization of the latter.

At the heart of the vessel planning problem lies a complex problem in combinatorial mathematics. There is currently a large scope for research into the theoretical and empirical properties of this most interesting problem.

Chapter 5

Conclusions and Future Work

Each of the preceding chapters has presented relevant conclusions and directions for further research. The purpose of this chapter is twofold. First, we explore the type of problems that will be relevant in the optimization of maritime transport and operations over the next decade. Secondly, we discuss the challenges that we are likely to face as we implement these models within a business context.

5.1 Opportunities through 2018

Below we discuss several opportunities that will become increasingly important over the following decade.

5.1.1 Better integration of maritime transport in the supply chain

Maritime transport plays a key role in the supply chains of several specific goods, such as automobiles, liquified natural gas, coal, petroleum, and chemicals. These sectors are particularly dependent on maritime transport because of the large volume or weight of the goods being transported, the distances involved, and the low value density of the products. In many of the aforementioned cases, ocean carriage is performed using specialized vessels that are designed exclusively for use in one industry. Furthermore, some of these products also require unique and expensive infrastructure at the loading and unloading points.

We anticipate important monetary benefits from a holistic analysis of the logistic chain

in each of these industries. One would seek to globally optimize facility location, vessel deployment, vessel routing, and shipment schedules in the presence of uncertain demand. Although this would appear to be an overly ambitious formulation, we should emphasize that the model would apply only to a very specific product, a small number of origin and destination points, and a fairly elaborate set of business rules, all of which would tend to make the models more tractable. We are currently aware of one such initiative targeted at the distribution of LNG.

We believe this idea would be particularly attractive to industrial shippers, as they exert control over both the production and the transportation components of the supply chain. Companies that ship their goods using a combination of owned and chartered vessels would benefit from such an initiative as well. In cases where shipment is made exclusively through charters, one can think of a consortium of charterers joining forces to adapt their practices in a way that generates added value to the supply chain as a whole. A key challenge in this regard is that of discovering the appropriate incentives for all members of the supply chain to cooperate for a greater good, a complex game-theoretical question which has received much attention in recent years.

5.1.2 Revenue management

Revenue management has been successfully used in the airline, hotel, rental car, retail, and energy sectors, amongst others (Talluri and Van Ryzin, 2004). The goal of revenue management is to establish pricing, selling quantities, and selling times to maximize revenue in the presence of uncertain demand and perishable capacity. Although revenue management is currently in use in some niche sectors within maritime transportation, namely cruise ships and ferries, we believe that this discipline could have a tremendous impact on the pricing of terminal services and containerized traffic.

One of the key elements of a revenue management strategy is the ability to segment the customer base according to their willingness to pay for the goods or services being offered. Thus far, the shipping industry has generally relied on the value of the merchandise being shipped as an indicator of willingness to pay. In microeconomic terms, we could consider

this a form of first degree price discrimination. There is an opportunity to segment the customer base along a second dimension, namely that of delivery speed. This would be implemented as second degree price discrimination, where each customer would select the price-service combination that suits her best. Customers that insist on a fast delivery will pay a premium for a direct shipment. On the other hand, customers that are not so keen on delivery speed will pay a lower price for a shipment that may be sent on a slower vessel, or through a longer chain of transshipment points. Thus far, oil has been relatively inexpensive, and the additional cost of a fast delivery was of little concern to many manufacturers. With rapidly increasing fuel costs, the tradeoff between delivery time and cost will gain much relevance.

Our informal inquiries have found that a few liner companies utilize revenue management today. However, the veil of secrecy that surrounds their business practice has prevented us from learning much about the scope of their capabilities. We do know of at least one major carrier that is actively pursuing a revenue management strategy at present. We believe these efforts will be redoubled over the next decade. Recall that the competitive advantage gained by American Airlines through its introduction of yield management around 1985 was sufficient to drive a smaller competitor to bankruptcy within two years. Other major carriers had to develop yield management competencies quickly in order to remain viable. We believe that early adopters of revenue management in the liner business will enjoy a substantial competitive advantage, and this will lead to a race to develop such capabilities over the next decade.

The doctoral dissertation of Maragos (1994) presents a simplified model of revenue management for the liner sector. Ting and Tzeng (2004) propose a slot allocation model for revenue management of container shipping. These two works demonstrate some of the particularities of the liner business that prevent us from replicating the systems developed in the airline industry directly.

A container ship is to an airplane what a container yard is to a hotel. Container terminals currently offer long term pricing contracts with incentives to attract large volumes from liners. The presence of a small number of clients and such long term contracts may impede

the introduction of yield management for terminals.

On the other hand, there are at least two important factors that will contribute to the push for yield management use in maritime terminals:

- As large stevedoring consortia acquire additional terminals, they become providers of a *network* service. In this case, the stevedoring companies control the nodes of the network, whereas a traditional revenue management interpretation focuses on the arc capacities and prices. A stevedoring company could balance work loads amongst its multiple terminals by adjusting the lift prices it offers to liner companies.
- The use of advanced terminal operating systems (TOS) at many terminals may facilitate the deployment of a yield management system. This is true because the TOS databases already contain much of the data that would be needed to generate fairly detailed demand forecasts (e.g. merchandise type, volume, date of shipment, origin and destination points, etc.). In this regard, the TOS providers stand to create a very profitable line of business by offering yield management engines as complements to their core products.

5.1.3 Focus on the environment

Waterborne transportation is, with regards to the emission of greenhouse gases, one of the most environmentally friendly methods of transport. Consider that 80% of all goods (by volume) are transported by ship, yet waterborne transportation generates half the greenhouse emissions generated by air transport (DNV Research and Innovation, 2007). As new environmental regulations are introduced (Jowit, 2008) and carbon trading schemes become available, the industry will place emphasis on reducing vessel emissions, both at port and while in transit.

On one hand, this environmental focus will lead to new engine, fuel, hull, anti-fouling, and propulsion designs. On the other hand, a new multi-objective approach to vessel routing and deployment will include the fleet's emissions in addition to traditional measures of cost and revenue. Emissions reduction will either be considered as a goal in its own right, or converted into an additional cost factor to be minimized. This latter approach is obviously

attractive in the presence of a viable market for carbon offsets. When additional limits for emissions (notably SO_x and NO_x) are written into law, our challenge will be to accurately capture the emissions footprint of the decision variables in our models.

Short sea shipping (SSS) has tremendous potential to reduce emissions and congestion in port cities by replacing much of the truck traffic that is now utilized to carry merchandise in and out of these ports. We envision the need for models to coordinate the transshipment between SSS vessels (consider the equivalent of a cross-docking strategy), as well as a multimodal component coordinating such vessels with trucks and trains arriving at the terminals.

In contrast to its relatively low greenhouse footprint, the shipping industry has the potential to generate extensive damage from oil and chemical spills. We envision new models to optimize the location and operation of containment crews. These models will consider the stochastic nature of the location, magnitude, and impact of future spills.

5.1.4 New patterns in world trade

Over the last three decades, China and South East Asia have become the world's manufacturers. As these countries transition to a new economic and political status, they will witness an increased standard of life, higher wages, and depletion of natural resources. In turn, these changes may motivate manufacturers to seek new opportunities in other countries in Asia, Africa, and South America. China and India will become powerful markets in their own right, and a larger proportion of their production will be targeted towards internal consumption. The shipping industry will have to adjust to these new patterns by redeploying vessels to the new trade routes. Initially, we foresee an increase in bulker traffic from Africa and South America to China and India. As the latter shift away from cheap manufacturing, the flow of raw materials will be reshaped, and a higher proportion of shipments will remain within Africa and South America.

It should be noted that the intensity of traffic between the Far East and North America or Europe has been a key driver of shipping demand over the past decades. These East-West routes are possibly the longest conceivable between major population blocks. As such, they

generate enormous demand for ton-miles of transport. If these long East-West routes were to be replaced with shorter East-West or South-North routes, the world fleet could easily outsize demand. Current projections of growth for total world tonnage may not consider the impact of this effect. In the past, an oversupply of vessels has led to a dramatic collapse in transport prices, which has in turn caused innumerable bankruptcies in the shipping sector (Stopford, 1997). Additionally, the optimal vessel design for a long East-West route may be suboptimal on shorter voyages, and many of the vessels that are now being designed and built for the longer trades may not find profitable employment if conditions change abruptly. We envision robust fleet sizing and deployment algorithms that would consider the uncertainties of the shipping market over a strategic planning horizon, and would aid shipping companies in making the best possible decisions under such circumstances.

5.1.5 Focus on safety

The risk of terrorist attacks will remain high over the next decade. Large maritime ports are particularly vulnerable to terrorist activity, either as important targets in their own right, or as entry points for weapons and personnel. This vulnerability follows from the enormous volume and variety of merchandise that is handled on a daily basis at hundreds of ports worldwide. Furthermore, many ports employ questionable security practices and are notorious foci of corruption. It is clearly impossible to check all merchandise against the presence of a terrorist device. Therefore, the development of statistical techniques and certification procedures to facilitate this task will become a paramount concern. Whereas several ports in the United States are already using such analytical tools, we envision growing demand for increasingly sophisticated statistical solutions.

5.1.6 Arctic exploration and the Northwest Passage

It is believed that the Arctic holds a quarter of the oil and gas reserves in the planet. As the polar caps recede, new opportunities for exploration will become economically attractive. The extraction and transportation of oil and gas under extreme weather conditions poses important problems in engineering and logistics design. Consider, for instance, that the waters around the extraction facilities are likely to be free of ice only intermittently throughout

the year. Transportation out of the facilities will therefore be subject to additional uncertainty, and provisions may be necessary to store production over extended periods of time. The vessels that operate in the Arctic will be highly specialized and will therefore require a substantial capital premium over regular oil and gas vessels. In order to operate these vessels profitably, a combined fleet may be necessary, with the specialized vessels transshipping their load to regular vessels at designated points. The problem of locating and sizing such transshipment and storage facilities will certainly require an optimization-based solution approach.

5.1.7 Speculation in the shipping sector

Over the last decade, we have witnessed two major economic crises built, to a large extent, on exuberant behavior: the dot-com crash of 2001, and the subprime housing crisis of 2007. In the first instance, telecommunications companies, based on overly optimistic forecasts, built fiber optic networks that would have accommodated several multiples of the existing demand. Not surprisingly, bandwidth prices crashed, and several companies were sold, in industry parlance, at pennies on the dollar. This event appears to have taken most investors and leaders in the telecommunications sector by surprise.

In the case of the subprime mortgage crisis, homeowners and mortgage lenders assumed that the double-digit appreciation of home values that was observed in many parts of the United States and Europe during the first half of the decade would persist. This led consumers to commit to monthly payments that they were unlikely to cover in the absence of a perpetual mechanism to extract additional funds from the *expected* appreciation of their homes. This time, several investors took note of the incoherence of this market, and made billions of dollars by creating and trading on appropriate financial instruments (Anderson, 2007).

Past behavior does not guarantee future behavior. The fact that the shipping sector (and in particular, container shipping) has grown at double-digit rates over the last decade does not mean it can continue to grow at this pace. We must question the fundamental assumptions behind the forecasts of growth in this sector. We must uncover every factor

that could impinge on the forecast before accepting it. Is there enough steel in the world to build so many vessels? Is there room in the existing ports and canals to handle the implied traffic? At the rate that oil prices are increasing today, will there be enough bunker fuel for all these vessels ten years from now? Today we are planning for 16,000 TEU vessels (Seatrade Asia online, 2007). Perhaps we have forgotten the Jahre Viking and the Batillus class of supertankers.

Any market that behaves with exuberance opens itself to the influence of short sellers. The shipping market may be prone to such exuberant behavior, given its history and affinity for risk. We believe that, for a team with the correct set of skills in finance and transportation management, large economic rewards may become apparent over the next decade.

5.2 The challenges

The implementation of advanced techniques from management science (and operations research in particular) to the maritime sector poses several significant challenges. In this section, we discuss some practical challenges that still need to be addressed by the operations research community.

5.2.1 The complexity of the models impedes their adoption

The models and algorithms employed in operations research are mathematically advanced, and thus intimidating or bewildering to most stakeholders. Many stakeholders are reluctant to yield control of their operations to an algorithm they can't fully understand. One of the key challenges ahead consists of creating a dialogue that permits all stakeholders to participate in the development of the tools and become comfortable with their adoption. Based on our prior experience, it could be argued that the operations research community would be wise to seek the cooperation of professional communicators in this regard.

5.2.2 User interfaces: less is more

Oftentimes, the developers of operations research-based decision support tools overplay their hand when designing the software's user interface. The intention is generally a noble one: to

provide a system that is as sophisticated as possible. The result is often counterproductive: the tool is overly complex, has a steep learning curve, and appears alien to all its intended users. In this sense, we have for several years advocated spreadsheet-driven optimization tools. To begin with, most of the data that the analysts will handle will come from a spreadsheet or a database. The model's output is again likely to be analyzed by the different stakeholders using spreadsheet software. Given that most analysts are at ease with this interface, it seems only natural to use it as a platform for the entire application.

5.2.3 The duration and cost of O.R. projects

Another challenge arises from the extended times and large investments that are required to develop a model from its inception to the point where a working tool can be released. Often, these times are in excess of eighteen months. It is not uncommon for such efforts to require several highly paid operations research specialists, as well as significant investments in computer hardware and software. Most small companies can't afford such investments. Few organizations have the patience and commitment to support the operations research team for the duration of such lengthy projects. The development of commercially available, off-the-shelf tools to support maritime companies remains an attractive business option. A key challenge in this regard will be the isolation of each core problem, as well as the development of components to permit the adaptation of the model to the circumstances of each company.

Here we run into something of a paradox. If optimization is to provide a unique competitive advantage, surely it can't be sold as a generic tool that applies equally to all companies. From a (very simplified) game-theoretical perspective, if all companies purchased such a product, none of them would gain a competitive advantage, yet all would have an additional business expense. Furthermore, the first customer will pay a high price while the software is debugged and tuned. The second customer will then acquire the finished software product, and as a byproduct, the knowledge gained by the software vendor during the first installation. The cost of this knowledge is unlikely to be properly represented in the second transaction.

In view of the above, the shipping companies would expect optimization tools to be highly customized. However, the cost of such customization is prohibitive for many smaller companies. We are aware of some software companies that have attempted to create products that can be highly customized at a lower cost. Their success so far is mixed, and additional insights are needed into this issue.

5.2.4 Uncertainty in the maritime sector

In the context of the strategic-level problems faced by the maritime sector, uncertainty can be so high as to derail the applicability of most models. Furthermore, there is typically no way to establish the distribution of underlying stochastic processes. In this sense, a new theory of robust optimization (Bertsimas and Weismantel, 2005) which does not require explicit knowledge of such distributions, should be of great interest to the maritime sector.

5.2.5 Underlying mathematical complexity

Finally, many of the problems faced by the maritime industry are extremely complex. The problems faced by container terminals are typically highly combinatorial and difficult to solve using commercially available software. Many models that attempt to represent multiple components of a supply chain are likely to become extraordinarily large and intractable. On the other end of the spectrum, an oversimplified model may lead to inaccurate, or fully incorrect conclusions. From the perspective of the problem data, Simchi-Levi, Kaminsky, and Simchi-Levi (2008) comment on the benefits of geographical aggregation to simplify logistics models. From the perspective of the algorithms, it is essential to attack every model using a wide variety of tools. We believe that hybrid optimization methods, which leverage the qualities of exact and heuristic methods, hold tremendous potential in the quest to address highly complex models.

We stress the need to focus on good, rather than optimal, solutions. The data that is available is typically subject to considerable uncertainty and inaccuracy. It is therefore pointless to insist on optimality.

In summary, although new mathematical insights will be essential to the solution of increasingly complex problems, the most important obstacles to the adoption of operations

research in the maritime industry are likely to be organizational in nature.

The models and algorithms that were presented in this dissertation are representative of the problems that are faced today by the maritime industry. As new challenges arise within the shipping industry, and as the sector becomes more open to advanced mathematical modeling techniques, we hope to be at the forefront of this exciting field.

Bibliography

- Abara, J., 1989. Applying integer linear programming to the fleet assignment problem. *Interfaces* 19, 20–28.
- Ahuja, R. K., Magnanti, T. L., Orlin, J. B., 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, New Jersey.
- Alderton, P., 2005. *Reeds Sea Transport Operations and Economics*, 5th Edition. Thomas Reed Publications.
- Alvarez, J., 2006. *Terminal Equipment in Asian Maritime Ports*. Working Paper, Universitat Pompeu Fabra.
- Anderson, J., March 9, 2007. *Winners amid gloom and doom*. The New York Times.
- Armacost, A. P., Barnhart, C., Ware, K. A., Wilson, A. M., 2004. UPS optimizes its air network. *Interfaces* 34 (1), 15–25.
- Bendall, H. B., Stent, A. F., 2001. A scheduling model for a high speed containership service: A hub and spoke short-sea application. *International Journal of Maritime Economics* 3, 262–277.
- Bertsekas, D. P., 1999. *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts.
- Bertsimas, D., Tsitsiklis, J., 1997. *Introduction to Linear Optimization*. Athena Scientific, Belmont, Massachusetts.
- Bertsimas, D., Weismantel, R., 2005. *Optimization Over Integers*. Dynamic Ideas.
- Bish, E. K., Chen, F. Y., Leong, Y. T., Nelson, B. L., Wing, J., Ng, C., Simchi-Levi, D., 2005. Dispatching vehicles in a mega container terminal. *OR Spectrum* 27.
- Böse, J., Reiners, T., Steenken, D., Voß, S., 2000. Vehicle dispatching at seaport container terminals using evolutionary algorithms. In: *Proceedings of the 33rd Hawaii International Conference on System Sciences*. IEEE, Piscataway, pp. 1–10.
- Carrotta, A., 2005. *Integrating combinatorial optimization into constraint programming: application to vehicle routing problems*. Ph.D. thesis, Università di Bologna.
- Castro, J., 2003. Solving difficult multicommodity problems with a specialized interior-point algorithm. *Annals of Operations Research* 124, 35–48.
- Cho, S.-C., Perakis, A., 1996. Optimal liner fleet routing strategies. *Maritime Policy and Management* 23, 249–259.
- Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D., 2007. Maritime transportation. In: Barnhart, C., Laporte, G. (Eds.), *Handbooks in OR & MS*. Vol. 14. Elsevier B.V, pp. 189–284.
- Christiansen, M., Fagerholt, K., Ronen, D., 2004. Ship routing and scheduling: Status and perspectives. *Transportation Science* 38, 1–18.

- Container Management Magazine, 2007. Container management magazine website, viewed December 2007. <http://www.container-mag.com/top120.php?SID=>.
- Cormen, T., Leiserson, C., Rivest, R., 1990. Introduction to Algorithms. The MIT Press, Cambridge, Massachusetts.
- Dijkstra, E., 1959. A note on two problems in connexion with graphs. *Numerische Mathematik* 1, 269–271.
- DNV Research and Innovation, 2007. Technology outlook 2015. Oslo, Norway.
- Fagerholt, K., 1999. Optimal fleet design in a ship routing problem. *Intl. Trans. in Op. Res.* 6, 453–464.
- Fagerholt, K., Lindstad, H., 2000. Optimal policies for maintaining a supply service in the Norwegian Sea. *Omega* 28, 269–275.
- Fagerholt, K., Lindstad, H., September 2007. Turborouter: An interactive optimisation-based decision support system for ship routing and scheduling. *Maritime Economics & Logistics* 9 (3), 214–233.
- Faststream UK, 2008. Faststream recruitment website. <http://www.faststream.co.uk/Shipping>.
- Fourer, R., Gay, D., Kernighan, B., 2003. AMPL, A Modeling Language for Mathematical Programming, 2nd Edition. Thomson.
- Glover, F., Kochenberger, G. A., 2003. Handbook of Metaheuristics. Kluwer Academic, Boston.
- Glover, F., Laguna, M., 1997. Tabu Search. Kluwer Academic, Boston.
- Gopalan, R., Talluri, K., 1998. Mathematical models in airline schedule planning: A survey. *Annals of Operations Research* 76, 155–185.
- Hapag-Lloyd, 2007. Hapag-Lloyd website, viewed December 2007. <http://www.hapag-lloyd.com>.
- ILOG S.A., 2003. ILOG CPLEX 9.0 Reference Manual. ILOG S.A., Mountain View, California.
- ILOG S.A., 2006. ILOG CPLEX 10.0 Reference Manual. ILOG S.A., Mountain View, California.
- Imai, A., Sasaki, K., Nishimura, E., Papadimitriou, S., 2006. Multi-objective simultaneous stowage and load planning for a container ship with container rehandle in yard stacks. *European Journal of Operational Research* 171, 373–389.
- International Maritime Organization, May 2006. <http://www.imo.org/home.asp>.
- Jones, D., Mirrazavi, S., Tamiz, M., 2002. Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European Journal of Operations Research* 137.
- Jowit, J., June 15 2008. Cargo ships told to go green by slowing down. *The Observer*, 11.

- Kim, K., Kang, J., Ryu, K., 2004. A beam search algorithm for the load sequencing of outbound containers in port container terminals. *OR Spectrum* 26, 93–116.
- Kim, K. H., 1997. Evaluation of the number of rehandles in container yards. *Computers and Industrial Engineering* 32.
- Kozan, E., 2000. Optimising container transfers at multimodal terminals. *Mathematical and Computer Modelling* 31.
- Lloyd's List, 2008. Lloyd's list jobs. <http://www.lloydslistjobs.com/l1/careers/index.htm>.
- Maragos, S. A., 1994. Yield management for the maritime industry. Ph.D. thesis, Massachusetts Institute of Technology.
- Mason, R., McKenney, J., Carlson, W., Copeland, D., 1997. Absolutely, positively operations research: The Federal Express story. *Interfaces* 27, 17–36.
- Montfort, A., Aguilar, J., Gómez-Ferrer, R., Arnau, E., Martínez, J., Monterde, N., Palomo, P., 2001. Terminales marítimas de contenedores: el desarrollo de la automatización. Instituto Portuario de Estudios y Cooperación de la Comunidad Valenciana.
- NIMA, 2001. Distances Between Ports, 11th Edition. National Imagery and Mapping Agency, Bethesda, Maryland.
- Notteboom, T. E., 2006. The time factor in liner shipping services. *Maritime Economics & Logistics* 6, 19–39.
- Powell, B. J., Perakis, A., 1997. Fleet deployment optimization for liner shipping: an integer programming model. *Maritime Policy and Management* 24, 183–192.
- Rana, K., Vickson, R., 1991. Routing container ships using Lagrangean relaxation and decomposition. *Transportation Science* 25, 201–214.
- Rosenthal, E., April 2008. Environmental cost of shipping groceries around the world. <http://www.nytimes.com/2008/04/26/business/worldbusiness/26food.html>.
- Rushmeier, R., Kontogiorgis, S., 1997. Advances in the optimization of airline fleet assignment. *Transportation Science* 31, 159–169.
- Seatrade Asia online, November 2007. Samsung unveils 16,000 teu design. <http://www.seatradeasia-online.com/News/1932.html>.
- Sherali, H., Al-Yakoob, S., 2006. Determining an optimal fleet mix and schedules: Part I – single source and destination. In: Karloff, J. (Ed.), *Integer Programming: Theory and Practice*. Taylor & Francis, Ch. 6, pp. 137–166.
- Sherali, H., Maguire, L., 2000. Determining rail fleet sizes for shipping automobiles. *Interfaces* 30, 80–90.
- Simchi-Levi, D., Chen, X., Braca, J., 2005. *The Logic of Logistics*. Springer.
- Simchi-Levi, D., Kaminsky, P., Simchi-Levi, E., 2008. *Designing and Managing the Supply Chain*. McGraw-Hill.

- SMDG, October 2000. UN/EDIFACT bayplan message MOVINS user manual (implementation guide). User Group for Shipping Lines and Container Terminals, 3rd Edition, <http://www.smdg.org/>.
- SMDG, February 2001. UN/EDIFACT bayplan message BAPLIE user manual (implementation guide). User Group for Shipping Lines and Container Terminals, 2nd Edition, <http://www.smdg.org/>.
- Song, D., Zhang, J., Carter, J., Field, T., Marshall, J., Polak, J., Schumacher, K., Sinha-Ray, P., Woods, J., January–March 2005. On cost-efficiency of the global container shipping network. *Maritime Policy & Management* 32 (1), 15–30.
- Spinnaker Consulting, L., 2008. Shippingjobs.com. <http://www.shippingjobs.com/jobseekers.asp>.
- Stahlbock, R., Voß, S., October 2007. Operations research at container terminals - a literature update. *OR Spectrum* 30 (1), 1–52.
- Steenken, D., Voß, S., Stahlbock, R., 2004. Container terminal operation and operations research – a classification and literature review. *OR Spectrum* 26, 3–49.
- Steenken, D., Winter, T., Zimmermann, U., 2001. Stowage and transport optimization in ship planning. In: Grötschel, M., Krumke, S. O., Rambau, J. (Eds.), *Online Optimization of Large Scale Systems*. Springer, Berlin, pp. 731–745.
- Stopford, M., 1997. *Maritime Economics*, 2nd Edition. Routledge, New York.
- Talluri, K., Van Ryzin, G. J., 2004. *The Theory and Practice of Revenue Management*. Kluwer Academic.
- The Wall Street Journal, June 2008. Markets data center. <http://online.wsj.com/mdc/public/page/mdccommodities.html>.
- Ting, S.-C., Tzeng, G.-H., 2004. An optimal containership slot allocation for liner shipping revenue management. *Maritime Policy & Management* 31, 199–211.
- UNESCAP, 2007. *Regional shipping and port development*. New York.
- Vanderbeck, F., Wolsey, L. A., 1996. An exact algorithm for IP column generation. *Operations Research Letters* 19, 151–159.
- Vis, I., de Koster, R., 2003. Transshipment of containers at a container terminal: An overview. *European Journal of Operations Research* 147, 1–16.
- Vis, I., Harika, I., 2004. Comparison of vehicle types at an automated container terminal. *OR Spectrum* 26, 117–143.
- White, R. D., Earnest, L., Jan 23, 2005. Congestion at cargo ports worldwide. *The Los Angeles Times*.
- Wolf, M., 2004. *Why globalization works*. Yale University Press.
- Zhang, C., Y. Wan, J. L., Linn, R., 2002. Dynamic crane deployment in container storage yards. *Transportation Research B* 36, 537–555.