Routing and Scheduling of Platform Supply Vessels

Case from the Brazilian Petroleum Industry

Written by:
Daniel Øyra Friedberg
Vidar Thune Uglane

Supervisor:
Henrik Andersson

Co-supervisors:
Vidar Gunnerud
Even Ambros Holte

Master's Thesis, Spring 2013

NTNU
Norwegian University of
Science and Technology

Preface

This Master's thesis is written in relation to our specialization within Applied Economics and Optimization at the Norwegian University of Science and Technology (NTNU). The thesis is the final assignment of our Master of Science degree at the Department of Industrial Economics and Technology Management.

The work of this Master's thesis started as a project for the Center for Integrated Operations in the Petroleum Industry during the summer of 2012 (Friedberg and Uglane, 2012a). For two months we were stationed at Petrobras and IBM Research in Rio de Janeiro, Brazil. This stay provided important preliminary data that is a vital part of our study. Furthermore, the work of this thesis is a continuation of a previous project thesis (Friedberg and Uglane, 2012b). In March 2013 we made a final trip to Brazil, where more recent data was gathered.

With regard to our recent visit to Petrobras Logistics' base in Macaé there are several people that deserve acknowledgment. We want to thank Gisele Lopes and Fernanda Cordeiro (E&P-SERV/US-LOG/GIOp) for supervising our stay in Macaé and providing an excellent overview of Logistics' operations. We would also like to direct thanks to Katia Andrade (E&P-SERV/US-LOG/CC), Ricardo Chagas (E&P-SERV/US-LOG/CC), Luiz Anderson (E&P-SERV/US-LOG/TM) and Bruno Vaz de Sampaio (E&P-SERV/US-LOG/OPRT).

Furthermore, we want to thank Mário Marcondes Machado and Romulo Loretti (E&P-ENGP/OPM/GIOp) for their assistance and expertise during our stay in Brazil. We also want to thank Paulo Viana (E&P-ENGP/OPM/GIOp) for showing interest in our work and providing us authorization to carry out our research.

With regard to guidance and feedback for the duration of this semester, we would like to thank our supervisor Associate Professor Henrik Andersson. He has been an invaluable resource to all questions regarding optimization. We also want to thank our co-supervisors Post Doctor Vidar Gunnerud at the Department of Engineering Cybernetics and Research Scientist Even Ambros Holte at MARINTEK for their insight and valuable input to this Master's thesis.

Trondheim, May 30th 2013

Daniel Øyra Friedberg

Vidar Thune Uglane

Summary

In this thesis we examine how Logistics in Petrobras may improve their decision making with regard to routing and scheduling of platform supply vessels (PSV). Petrobras are aiming to double their current production level by 2020. The expansion requires an increase in capacity on all levels of the organization. This includes Logistics, which will have to cope with the future expansion in addition to facing some underlying issues that already exist in their supply chain.

The logistics base for Campos basin is located in the city Macaé. Currently the port in Macaé is operating at maximum capacity and struggles to cope with increasing demand inquiries. We argue that an operational planning tool for routing and scheduling may contribute to more efficient order handling.

The thesis introduces the platform supply vessel routing and scheduling problem with refueling tankers (PSVRSP-RT), which consists of identifying optimal routes and schedules for the daily departures of PSVs servicing a set of offshore installations. The problem considered is formulated as a deterministic cost minimization problem and solved as a mixed integer linear programming problem (MILP).

For a given planning horizon the model intends to properly allocate a set of pickup and delivery orders to a fleet of PSVs that has one available departure per day. Among other tasks, the model solves the optimal choice of vessel type for each departure, delivery sequence of orders and optimal deployment locations for refueling tankers. The objective function minimizes fixed and variable sailing costs, in addition to minimizing the penalty costs of delay and express delivery of orders.

Friedberg and Uglane (2012b) provided an arc-flow formulation of the original PSVRSP. In the Master's thesis the same model is expanded by including the aspect of refueling tankers. According to the reviewed literature, there seems to be a clear trend towards the use of column generation to solve maritime ship routing problems. The arc-flow model is therefore decomposed using the Dantzig-Wolfe decomposition (DWD) approach, which results in a path-flow model. In the decomposed model, a subproblem (SP) is solved for each combination of departures and vessels, whereas the restricted master problem (RMP) incorporates the common constraints and the pool of columns generated in the subproblems. Column generation is used to solve the linear programming relaxation of the MILP problem. Furthermore, Branch and Price (B&P) is implemented to close the MILP gap and provide an optimal integer solution.

A computational study tests a set of different approaches to column generation and other aspects of the B&P tree to speed up the solution process for the path-flow model. Testing reveals that the break-first-then-optimum (BFO) procedure is the most promising column generation approach. After implementing a satisfactory B&P scheme, the path-flow model is compared with the arc-flow model for several problem instances. Results indicate that the arc-flow model performs better for small problem instances, but as the size of the problems increases the path-flow model becomes superior. An advantage of column generation is that the model is able to provide decent bounds within a reasonable time frame, despite experiencing a tail-off effect in the B&P tree.

The model is applied to a base case from Campos basin to evaluate the problem from an economical perspective. In terms of economical value, knowing how the model is affected by variations in input parameters adds value on a strategic level. With regard to the PSV fleet size and mix it is evident that a balanced composition is important in terms of cost minimization. The results reveal that a scenario involving excess capacity, i.e. a majority of PSV 3000 vessels, causes a significant increase in direct costs. In contrast, the instances involving a capacity deficit reveal a potential increase in indirect costs related to delay and express delivery costs.

When we alter parameters for the offshore refueling operations, the results present only small deviations in total costs. In fact, the results indicate that the buoys for anchoring refueling tankers are properly positioned. Changing the deployment locations of refueling tankers to non-optimal buoys make little difference in terms of direct costs. With regard to the order prioritization system, the analysis provides an important intuition about the sensitivity of the delay penalty. The direct costs, which consist of sailing distance and fixed leasing rates for PSVs, are not significantly influenced by large alterations in delay penalty.

Logistics face some issues that must be addressed before the model may be implemented in practice. Internal uncertainty factors arise due to lack of transparency, capacity restrictions and stakeholders' mistrust in logistics operations. To increase trust in the system by more reliability in order handling, Logistics have introduced fixed routes and schedules for PSVs. In relation to the Petrobras 2020 vision and the development in Santos basin, the future state of Logistics may become more adaptable to a dynamic planning environment. As Santos basin is under development, drilling rigs and other non-stationary installations will be the main source for demand. Fixed routes would therefore have to be reset frequently. Furthermore, as distances from the depot increase, refueling offshore might in fact become a necessity to increase the range of the PSVs.

Sammendrag

I denne masteroppgaven undersøker vi hvorvidt logistikkavdelingen (Logistics) i Petrobras kan forbedre sin beslutningstaking med tanke på ruteplanlegging for supplyfartøy (PSV). Petrobras har som mål å doble sitt nåværende produksjonsnivå innen 2020. En slik ekspansjon krever økning i kapasitet på alle nivåer i organisasjonen, inkludert Logistics. Logistics kjenner til en del problemer med sin eksisterende verdikjede, noe som kan by på utfordringer i forhold til de store ekspansjonsplanene til selskapet.

Logistikkbasen til Campos-bassenget utenfor kysten av Brasil ligger i byen Macaé. Havnen i Macaé benytter nå all tilgjengelig kapasitet og sliter med å møte økende etterspørsel. Vi argumenterer for at et operasjonelt ruteplanleggingsverktøy kan bidra til mer effektiv håndtering av etterspørselen.

Oppgaven introduserer ruteplanleggingsproblemet for supplyfartøy som inkluderer påfyllingstankere for drivstoff (PSVRSP-RT). Problemet består av å finne optimale ruter for daglige avganger av PSVer som betjener et sett med offshoreinstallasjoner. Vi formulerer problemet som et deterministisk kostnadsminimeringsproblem og løser det som et blandet heltallsproblem.

For en gitt planleggingshorisont forsøker den matematiske modellen å allokere et sett med ordre, både for leveranse og henting, til en flåte av PSVer som har en avgang per dag. Modellen finner den optimale fartøystypen til hver avgang, ordrebehandlingssekvensen på hver plattform, samt optimal plassering av påfyllingstankere. Målfunksjonen minimerer faste og variable seilekostnader. I tillegg minimerer den straffekostnaden ved sen ordrebehandling, samt kostnaden ved å sende ekspressfartøy for å betjene ordrer.

Friedberg og Uglane (2012b) presenterte en arc-flow-formulering til det originale PSVRSP. I masteroppgaven er den samme modellen utvidet til å inkludere påfyllingstankere. Det viser seg etter en gjennomgang av litteratur at det en er klar trend mot å benytte seg av kolonnegenerering til å løse maritime ruteplanleggingsproblemer. Vi dekomponerer derfor arc-flow-modellen ved bruk av Dantzig-Wolfe-dekomponering, noe som resulterer i en path-flow-modell. I den dekomponerte modellen blir subproblemer løst for alle kombinasjoner av fartøystyper og avganger. Det begrensede masterproblemet inneholder alle fellesrestriksjoner, samt de kolonnene som blir generert i subproblemene. Kolonnegenerering blir brukt til å løse den lineære relakseringen av heltallsproblemet. Branch and Price (B&P) implementeres for å lukke dualitetsgapet som ofte oppstår, samt sørge for at en optimal heltallsløsning blir funnet.

Vi tester forskjellige fremgangsmåter til kolonnegenereringsprosessen i path-flow-modellen, i tillegg til andre aspekter ved B&P-algoritmen, for å korte ned løsningstiden. Testene viser at break-first-then-optimum (BFO) er den mest lovende kolonnegenereringsmetoden. Etter at vi har implementert en tilfredsstillende B&P-algoritme sammenligner vi path-flow-modellen med arc-flow-modellen for flere probleminstanser. Sammenligningen viser at arc-flow-modellen gir best resultater på de minste instansene, mens path-flow-modellen utvilsomt gir bedre resultater når størrelsen øker. En fordel med path-flow-modellen er at den raskt reduserer dualitetsgapet, mens en ulempe er at den har problemer med å lukke gapet fullstendig på grunn av tail-off-effekten som oppstår.

Vi gjør videre noen økonomiske analyser med modellen på en instans fra Campos-bassenget. Det kan være verdifullt i et lenger planleggingsperspektiv å vite hvordan forandringer i parametere påvirker modellens løsninger. Det viser seg at en balansert flåte av PSVer er viktig for å minimere kostnader. Resultatene viser at en instans der for mye kapasitet, i form av for mange PSV 3000 i flåten, fører til store økninger i direkte kostnader. For lav kapasitet fører til en økning i indirekte kostnader i form av forsinkelser, samt kostnader relatert til leveranse av ordre med ekspressfartøy.

Når vi endrer parameterne som har med påfyllingstankere å gjøre, viser resultatene små forandringer i totalkostnader. De indikerer at bøyene der tankerne kan forankres er posisjonert relativt bra. De direkte kostnadene påvirkes lite av å endre en påfyllingstankers posisjon til en ikke-optimal bøye. I forhold til ordreprioriteringssystemet viser analysen et viktig aspekt med tanke på sensitiviteten til straffekostnaden ved forsinkelser. De direkte kostnadene, som består av PSVenes variable og faste kostnader, blir ikke betydelig påvirket av endringer i straffekostnaden.

Logistics står overfor en del problemer som må håndteres før modellen kan implementeres i praksis. Interne usikkerhetsfaktorer oppstår grunnet mangel på åpenhet, kapasitetsrestriksjoner og andre avdelingers mistillit til logistikkprosessene. Logistics har introdusert faste ruter for PSVer som et ledd i å forbedre samarbeidet med andre avdelinger, for dermed å øke forutsigbarheten i ordrebehandlingen. I forhold til Petrobras sine 2020-mål og utviklingene som skjer i Santos-området, mener vi at fremtidens Logistics kan være mer egnet til å ta i bruk et dynamisk ruteplanleggingsverktøy. Når Santos-bassenget utvikles vil borerigger og andre mobile installasjoner stå for mesteparten av etterspørselen. Faste ruter vil da være gjenstand for hyppige endringer. I tillegg vil distansene til oljefeltene øke, og påfylling offshore kan til og med bli en nødvendighet for å øke rekkevidden til PSVene.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This opening chapter provides a basis with regard to background information for the platform supply vessel routing and scheduling problem in Petrobras Logistics. Petroleum products are the most significant energy resources today, and account for approximately 37 % of the world's total energy consumption (World's Energy Resources and Consumption, 2010). It seems evident that the only way we can satisfy future energy consumption is in fact by maintaining, or even increasing, the oil production levels and constantly strive to provide more energy efficient solutions (Goodman, 2012).

Section 1.1 provides information about Petrobras, their vision and the Brazilian oil industry in general. In Section 1.2 the Logistics unit in Petrobras is presented in some detail and Section 1.3 describes the motivation and purpose of the thesis.

1.1 Petrobras 2020

Petrobras is the largest company in Latin America and was ranked the third most valuable energy company in the world in 2010 (Business Insider, 2012). The oil production is divided geographically based on basins spread along the Brazilian coastline. Figure 1.1 shows the area division of respective basins. Each basin is operated by an operations unit, which is administrating all processes within the boundaries of that basin.

The northern basin is called Espiritos Santos (ES in Figure 1.1), with headquarters located in the city Vitória. For the area surrounding Rio de Janeiro there are two basins, Campos basin (BC) and Rio basin (RIO). Due to the close proximity of BC and RIO, we will in the remainder of this report only refer to Campos. In the south the area is called Santos basin (BS). Santos is the most promising area with regard to future exploration and production. In this thesis we focus our attention mostly on Campos basin. The respective oil fields are located within each basin and often referred to as assets. Each asset may have several offshore production facilities located within its proximity. On occasion the location and division of facilities vary among

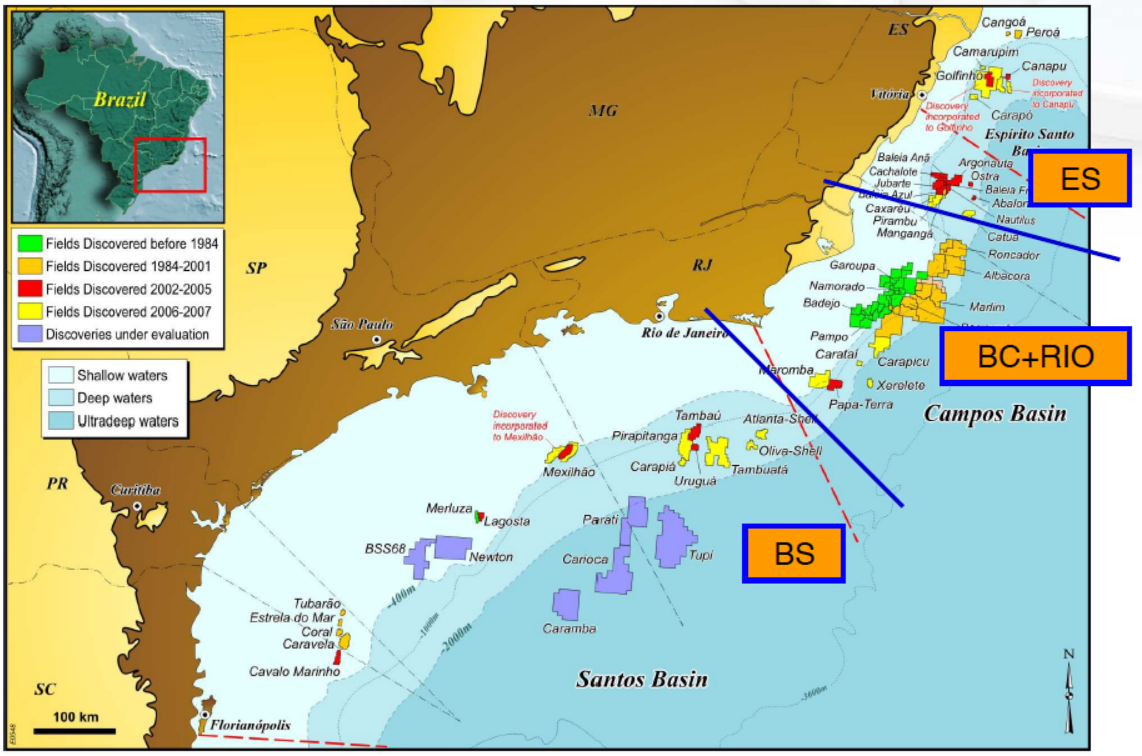

Figure 1.1: *Geographical division of basins*

the assets, meaning that they are not fully stationary. By production facilities we often refer to floating production storage and offloading units (FPSO), and traditional production platforms. From this point on we use the declarations production facility, installation and platform interchangeably.

The largest and oldest basin is Campos which accounts for about 80% of the Brazilian oil production (The Financialist, 2013). Some of the reservoirs in Campos are experiencing a decrease in production rate (Fraga et al., 2003). This has led Petrobras to focus more of their attention on the pre-salt fields further off the coast of Brazil, for the most part within the Santos basin area.

The pre-salt fields are located far offshore, in some instances more than 300 kilometers. Furthermore, the sea depth is usually extreme in these areas. This is illustrated by the Tupi field, one of the largest discoveries so far (see Figure 1.2). The water depth in these areas is usually around 2000-3000 meters, with an additional 3000-4000 meters of drilling to reach the reservoir. As a reference, the production on the Norwegian continental shelf has a distance to shore of approximately 200 kilometers with an average water depth of about 90 meters. Drilling depth in the North Sea is usually around 2000-3000 meters (Ducrotoy et al., 2000).

Historically Petrobras has experienced a rapid growth in production rate. The company went from producing 1 million barrels a day in 1997 to producing 2 million barrels in 2003. Due to the steady increase it led Brazil to be a self-sufficient producer of oil in 2006. Furthermore, Petrobras are aiming to double the current production by 2020 (Fried, 2011). The 2020 vision is to become one of the five largest integrated en-

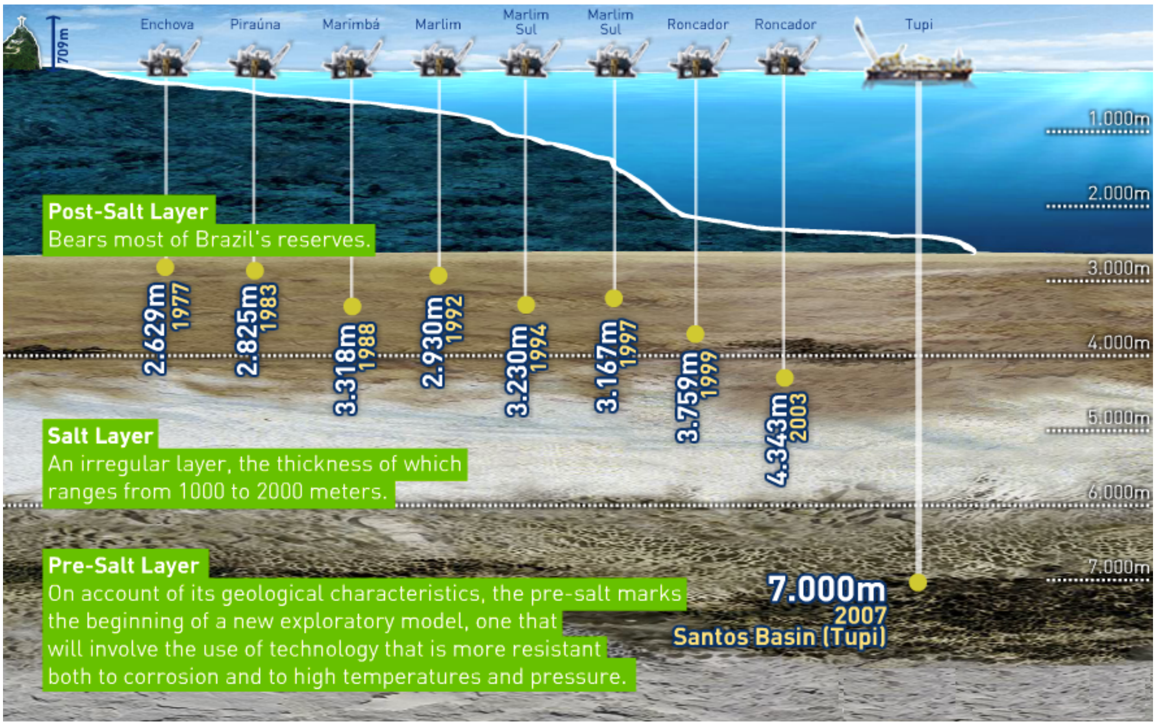

Figure 1.2: *Water depth comparison for Brazilian oil fields (Petrobras Vision for 2020, 2013)*

ergy companies in the world and the preferred choice among stakeholders (Petrobras Vision for 2020, 2013). To reach these goals the company will have to undergo a large increase in production while simultaneously improving the integration of processes across the supply chain. Among the service stakeholders that need to adapt to the arising challenges, is Logistics. This thesis focuses on challenges related to routing and scheduling of platform supply vessels within Logistics.

1.2 Logistics in Petrobras

All logistics tasks related to Campos basin are administrated from Macaé, a small city located approximately 200 kilometers north of Rio de Janeiro. The main task of Logistics is to manage resources between warehouses, suppliers, port and offshore facilities. In that sense, they are responsible for both land-based and offshore transportation of cargo. There is only one port available to serve Logistics' tasks within Campos basin, and it is located in Macaé. The type of cargo delivered varies in size and purpose of use, e.g. equipment, pipes, common commodities or bulk cargo. The Logistics unit in Petrobras is referred to as a service unit (US) and the formal term used is US-LOG. Figure 1.3 shows our interpretation of the flow of cargo within Logistics. The abbreviations in the figure are the actual terms used within the company and therefore based on Portuguese spelling.

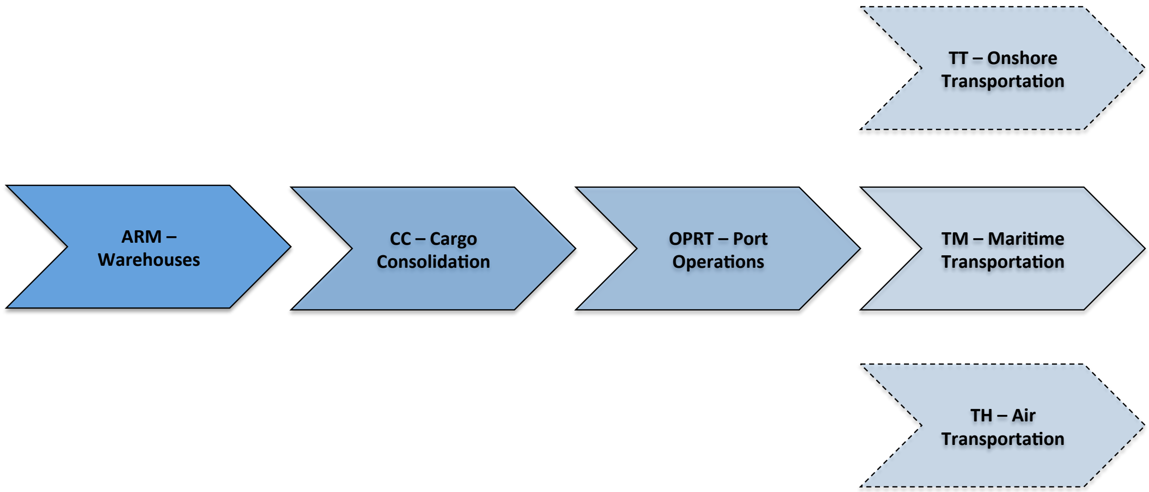

Figure 1.3: *Illustration of the supply chain in US-LOG*

Warehouses - ARM

Most warehouses managed by Logistics are located within close proximity in Macaé. They are responsible for keeping track of inventory and respond to cargo requests from offshore installations. Figure 1.4 shows the location of the warehouse sites in Macaé. The main warehouse is called Parque de Tubos (PT). PT is the most important warehouse in terms of total storage area, as well as being responsible for administrative tasks. In fact, PT is the only checkpoint used for verification of items. Hence, all materials and items must go through PT to be inspected before being shipped to other warehouse sites or the port. Due to the complicated infrastructure in Macaé, the warehouse locations may constitute possible bottlenecks in the supply chain.

Cargo consolidation - CC

Cargo consolidation (CC) are responsible for administering the cargo at the warehouses. These operations are located at the main warehouse, PT. The ARM personnel respond to requests made by CC, and prepares items for transit to the port or other locations. In general, CC are responsible for deciding transportation mode, size, destination and priority of the cargo. ARM carry out the order requests by unitization of items onto cargo containers and loading the cargo onto trucks.

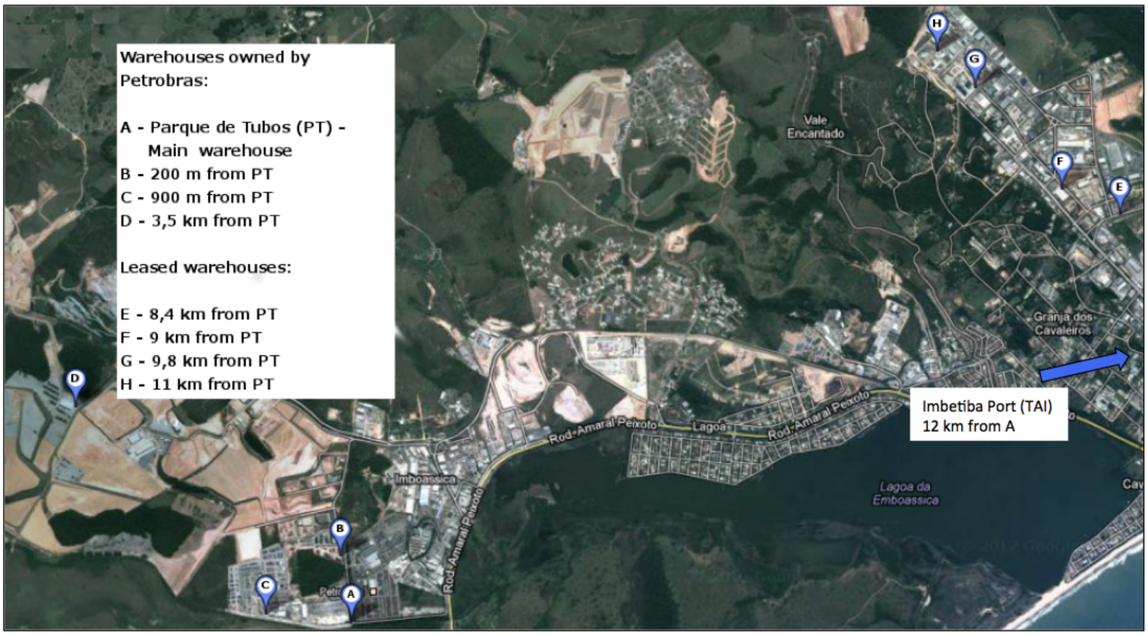

Figure 1.4: *Warehouse locations in Macaé*

The activities of CC are relevant to PSV routing and scheduling because they involve how platforms are prioritized. This was one of the main topics discussed in Friedberg and Uglane (2012b). CC are responsible for designing the schedules and routes used for shipping cargo. A major difficulty CC is facing is how to distinguish between normal and emergency order requests.

Port operations - OPRT

Port operations (OPRT) are responsible for all tasks upon cargo arrival from warehouses and loading/offloading activities within the harbour proximity. OPRT makes use of cranes, trucks and trailers to conduct the everyday tasks. Figure 1.5 shows the port terminal in Macaé which is owned and operated by Petrobras. The formal name of the port is Terminal Alfandegado de Imbetiba (TAI).

Maritime transportation - TM

Maritime transportation (TM) are responsible for monitoring the activities of vessels. Due to a monitoring system they are able to track the movement of the vessels and receive feeds on loading operations offshore. Currently the system is used to safeguard operations, but future use could potentially move towards traffic coordination, e.g. coordinating the speed of vessels to reduce fuel consumption.

Figure 1.5: *Terminal Alfandegado de Imbetiba (TAI)*

Figure 1.6 shows a screen shot from the monitoring system with an overview of Petrobras facilities located in a specific area of Campos basin. The nodes marked yellow are buoys for anchoring refueling tankers, while the rest are mostly production platforms and FPSOs. Today, TM have limited involvement in the daily vessel operations. Thus, most of the communication between the vessels and TAI is handled by OPRT, and the vessel arrival offshore is coordinated directly between the vessel and the facility. Although TM currently are less involved in the supervision of vessel operations, they are still worth mentioning as a potential participant in the future state.

Figure 1.6: *A selection of offshore facilities in Campos basin*

Platform supply vessels

Platform supply vessels (PSV) are utilized to supply cargo for offshore installations. Features of the PSVs vary depending on type of supply and company requirements for capacity and speed. Some PSVs are for instance used as tankers to transfer diesel oil to platforms. These vessels are often called oilers. PSVs may carry supply either on deck or in storage compartments below deck. Containers, pipes and other equipment is stowed on deck. Much of the platform orders are categorized as so-called bulk cargo. The bulk can be distinguished as dry bulk and liquid bulk. Pulverized cement (dry bulk), fuel (liquid bulk), water and chemicals are types of bulk cargo that are loaded in separate storage compartments below deck.

The PSVs are usually classified by dead weight tons, whereas 1500, 3000 and 4500 tons are most common in Petrobras. The PSV 4500 is mainly utilized for shipping commodity goods that feature standardized schedules, while PSV 1500 and PSV 3000 are used to serve more fluctuating demand. The focus in this thesis is on serving the fluctuating demand, therefore PSV 1500 and PSV 3000 are of particular interest. Most of the PSVs are leased by external companies and feature daily rates. The daily rates are approximately $ 25,000 and $ 35,000 for PSV 1500 and PSV 3000, respectively

In Table 1.1 some key features of the PSV 1500 and 3000 are listed. The information is taken from CBO, a Brazilian offshore shipping company of maritime support (Companhia Brasileira de Offshore, 2012). There is a wide variety of specifications for a PSV class. The table provides some features that are representative for PSV 1500 and 3000. Note that these numbers represent two specific vessels within the CBO fleet, i.e. CBO Macaé and CBO Campos.

Table 1.1: *Main characteristics of the PSV 1500 and the PSV 3000 (Companhia Brasileira de Offshore, 2012)*

Vessel type	PSV 1500	PSV 3000
Vessel name	CBO Macaé	CBO Campos
Length overall	63.95 m	67.00 m
Deadweight capacity	1460 t	3050 t
Service speed	10 knots	13 knots
Deck load	500 t	1500 t
Free deck area	240 m^2	620 m^2
Accomodations	18 persons	22 persons

Refueling tankers

Another important part of offshore operations is the use of refueling vessels. These vessels are tankers and are usually anchored at pre-specified locations. Petrobras currently have seven so-called refueling buoys, whereas six of them are located in

Campos basin. The buoys are possible anchoring positions for the tankers. Furthermore, Petrobras currently have a fleet of four tankers that are utilized for refueling operations.

Once the tankers are deployed they stay at the given buoy until the vessel is depleted and must return to shore for replenishment. Thus, the final allocation of these tankers has a strategic significance for operations offshore. In general the decisive factors in deciding the locations rely on weather conditions and demand in the area.

The main function of the refueling tankers is to provide diesel oil for platforms. In this case PSV oilers are used to transfer diesel between the refueling tankers and the platform. An average oiler has a capacity of carrying about 2,000 m^3 diesel oil, while the total capacity of a refueling tanker is approximately 18,000 m^3. The refueling tankers are also used to refuel other vessels, such as PSVs used for cargo transportation. The PSVs are rarely in the need of additional fuel to complete a trip, but the PSVs may still refuel offshore in order to save time at Imbetiba port. On average, a cargo PSV refuels about 200-300 m^3 diesel oil for its own consumption. The refueling process usually requires 4-5 hours, including docking and fueling operations.

1.3 Motivation and purpose

In this thesis a mathematical model is designed for PSV route planning and scheduling in US-LOG. We present an operational optimization model aimed for daily route planning, which requires short runtimes. Therefore much emphasis is put on technical aspects to reduce computational requirements. Different solution approaches are evaluated based on a review of existing literature. Furthermore, we utilize the model to make economical analyses on a scenario that arises from Campos basin.

Our most important research contributions are firstly the design, reformulation and implementation of a mathematical model that finds optimal routes and schedules for PSVs. Secondly, we include the positioning of offshore refueling tankers in the mathematical model, which we believe is a new aspect to existing routing models found in the literature. Thirdly, we conduct an analysis on the impact of varying input parameters to a real world scenario to prove the model's value as a decision support tool.

Although the model is designed as an operational planning tool, there might be other areas of usage. In relation to planning horizons, Christiansen et al. (2005) classify maritime transportation planning in the traditional way in terms of strategic, tactical and operational problems, with some examples shown in Table 1.2. From a supply chain perspective Simchi-Levi et al. (2009) define strategic level decisions as having long-lasting effect on the firm with a time frame greater than one year. Tactical level decisions are typically updated anywhere between once every quarter and once every year, while operational level refers to day-to-day decisions.

From Table 1.2 we see that most maritime routing and scheduling problems in the literature are in fact tactical. However, we argue that for PSVs the best approach to

Table 1.2: *Classification of optimization problems (Christiansen et al., 2005)*

Strategic (2 years -)	Tactical (3 months - 2 years)	Operational (days - 3 months)
Market and trade selection	Adjustment to fleet size and mix	Cruising speed selection
Ship design	Fleet deployment	Ship loading
Fleet size and mix	Routing and scheduling	Environmental routing
Port location, size and design	Berth and crane scheduling	
	Container stowage planning	

route planning is from an operational perspective. Many maritime problems cover long distances, which require vessels to sail voyages that may last up to several months. For the PSV routing and scheduling problem there are daily departures, and voyages typically last only a few days. In that sense we believe PSV routing resembles onshore routing problems, which typically cover the day-to-day routing of trucks, and thus are highly operational.

Today, US-LOG makes a tactical approach to route planning by fixing routes for a long period of time. The motivation for these fixed routes is to build trust and predictability in the underlying supply chain, which has proven a long lasting problem for Logistics. Petrobras suffers from frequent no-shows at the port, meaning vessels have to wait for the cargo to arrive. Furthermore, the large amount of so-called emergency requests is misused by platforms to secure on-time delivery. Thus, improving predictability through fixed routes might be a necessary approach in the current state.

In this thesis we will focus on optimizing a dynamic route generation process. Formerly, Logistics conducted route planning by dynamically generating routes, however without optimization. The current fixed routes are designed to cope with peaks in demand, which normally implies spare capacity onboard vessels. With the underlying processes in place, demand can be allocated to dynamically created routes in order to more efficiently utilize the fleet of PSVs. This thesis intends to incorporate realistic decision variables, parameters and constraints and thereby simulate dynamic route planning, which may become a possible future state for Petrobras. Furthermore, the economical study intends to utilize the operational planning tool for more long term decision making in Campos basin.

In Chapter 2 the problem description for the platform supply vessel routing and scheduling problem is presented. Chapter 3 gives a literature review of similar problems and some of their solution methods. In Chapter 4 the mathematical arc-flow formulation of the routing problem is explained in full. Chapter 5 presents the concept of Dantzig-Wolfe decomposition and decomposes the arc-flow formulation to a path-flow formulation. Chapter 6 introduces Branch and Price as a solution method for the path-flow model. In Chapter 7 and Chapter 8 the implementation with commercial software and computational results are laid out, respectively. The computational study is divided in two main parts; an evaluation based on technical aspects of the arc-flow and path-flow models and a study focusing on economical aspects. Furthermore, we give an overview of challenges when implementing an operational model in practice. Chapter 9 and 10 conclude our findings and discuss potential future research in light these findings.

Chapter 2

Problem Description

In this chapter we present the platform supply vessel routing and scheduling problem with refueling tankers in detail. The problem is to assign routes and schedules based on orders from offshore installations in Campos basin.

The PSVs begin and end their journey at the common depot in Macaé, i.e. Imbetiba port. The planning period is up to one week and the number of departures considered equals the number of days in the planning period. For a given planning period there is a fixed departure time every day. Each departure is assigned a single vessel.

The facilities considered are mostly production platforms, thus all facility positions are assumed to be stationary for the duration of the planning horizon. The facilities request delivery and pickup of cargo through an ordering system. Deck capacity restrictions on the platforms are not considered. Each order consists of a lower and upper time window for service, priority and capacity requirements on board a vessel. The lower time window for servicing an order is fixed. Thus, service cannot start earlier than the specified lower time window. Vessels may wait at the offshore facilities until the time window starts, and several orders may be serviced upon a visit. Service time required offshore is specified for each order and depends on the size of the order. The service time includes loading, offloading and docking operations offshore.

Only one visit per platform is allowed on a voyage. Due to capacity limitations in the port, the upper time window is soft. Service starting later than the upper time window incurs a lateness cost. To avoid orders with low priority being delayed indefinitely, there is a maximum delay for all orders. Orders that violate the maximum delay must be serviced by an express departure, which is available at any time. Thus, the express departures are not limited by a fixed schedule.

Refueling buoys are placed at strategic positions offshore. Furthermore, a fixed number of refueling tankers is available for anchoring at the specified buoy locations at the start of the planning period. Once a location is chosen a tanker is fixed in this position for the duration of the planning period. All PSVs are required to visit a tanker during a voyage, which will induce a fixed time for refueling. This requirement is a way of simulating the idea of refueling offshore and saving time in port later. This does not have a replenishment effect for PSVs, but it affects the routing aspect.

PSVs start at the common depot and visit one or more platforms and a refueling tanker before returning to the depot. The number of vessels of each type available within the planning horizon is predetermined. There is a limit on total sailing distance for each vessel type, and a limit on total duration of a voyage. The vessels are assumed to travel at constant cruising speed, which is similar for all vessels. Capacity varies depending on vessel type. The binding capacity constraint onboard a vessel is the deck space (m^2). Therefore each order is given accordingly. Bulk commodities are not considered in the problem.

The objective is to minimize costs of servicing orders from the platforms. Costs include fixed costs for a vessel, sailing costs of the voyages and lateness costs for cargo that is delivered overdue. Lateness costs are calculated based on the priority of the cargo that is delivered overdue. This cost is proportional to the importance of the cargo specified in the order. In addition, there is an express departure cost at the end of the planning horizon for each order that is not serviced by regular departures. The express cost is set high relative to other costs to ensure that orders are primarily allocated on regular departures.

Chapter 3

Literature Overview

This chapter provides an overview of existing literature related to the PSV routing and scheduling problem with refueling tankers, in light of the problem description.

A formal description of the PSVRSP-RT is a vehicle routing problem with pickup and delivery, soft time windows and intermediate refueling tankers (VRPPDSTWIRT). We base this description on literature reviewed in this chapter and the related study (Friedberg and Uglane, 2012b). For simplicity we only refer to the problem as the PSVRSP-RT.

In Chapter 1 we argued that the PSVRSP-RT resembles onshore vehicle routing problems in terms of planning horizon. The intention of Section 3.1 is to provide a review of related literature without particular regard to the mode of transportation. We then shift focus to literature more closely related to PSV routing and scheduling in particular, to get an understanding of where this topic stands within optimization. Solution methods are particularly important for the work in this thesis. Various methods are reviewed in light of the problems we find closely related to the PSVRSP-RT.

Section 3.1 gives a review of various VRPs that are relevant to the PSVRSP-RT, and provides an explanation to the formal description given above. In Section 3.2 an overview of articles specifically centered on the topic of routing and scheduling of platform supply vessels is provided. Lastly, Section 3.3 presents various solution methods used in related routing problems.

3.1 Vehicle routing problems

In general there are many similarities between maritime and road-based transportation problems. Unfortunately there has been done less work on maritime transportation problems compared to the classical vehicle routing problems (VRP). Christiansen (1996) presents a link between classical routing problems and their appearance as subproblems in maritime transportation problems. We start this section by going through some basic VRPs, before considering extensions of the VRP that relate to the problem described in this thesis.

Toth and Vigo (2002) present a collection of work done on the VRP. A more recent collection of articles can be found in Golden et al. (2008). Onshore routing problems are often assumed to contain a homogeneous fleet of vehicles. Ships on the other hand normally vary significantly in specifications. Most maritime models therefore assume a heterogeneous fleet. Routing of a heterogeneous fleet, and related work done on this subject, is described in Golden et al. (1984) and Baldacci et al. (2008).

An extension of the VRP includes time windows for delivery, which is described by Savelsbergh (1985) and Toth and Vigo (2002). In the vehicle routing problem with time windows (VRPTW) each customer has a time window constraint that must be met. Another interesting extension of the VRP that considers pickup (backhauls) in addition to delivery (linehauls) at each customer location can be found in Goetschalckx and Jacobs-Blecha (1989), Toth and Vigo (2002) and Deif and Bodin (1984). The vehicle routing problem with backhauls (VRPB) is focused on truck routing. Truck characteristics imply that loading can only take place in the rear end of the vehicle. This means that all linehaul customers in the VRPB must be visited before backhauls can be loaded onto the truck, i.e. all delivery before pickup. The deck of a platform supply vessel is loaded vertically by a crane and does however not have this limitation.

Related to the soft time windows that were introduced in Chapter 2, Fagerholt (2001) considers the multi-ship pickup and delivery problem with soft time windows (m-PDPSTW). In addition to an inner time window there exists an outer time window that is absolute. Violations of the inner time window induces a penalty cost in the objective function, whereas violation above the outer time window is not allowed. Fagerholt's motivation for introducing soft time windows is to obtain better schedules and reduce transportation costs. However, the motivation for the PSVRSP-RT is the port capacity limitations that may make it impossible to find feasible solutions with hard time windows.

Christiansen (1996) presents a generalization of the VRPTW; the pickup and delivery problem with time windows (PDPTW). In this problem each item has a specific origin and destination node. The model may be used in for instance container shipping, where items represent cargoes with ports as their origin and destination. Furthermore, the VRP with pickup and delivery and time windows (VRPPDTW) described by Desaulniers et al. (2002) includes a fleet of vehicles based at multiple terminals. In this case, each transportation request has a specified origin and destination. A special case of the VRPPDPTW is called the one-to-many-to-one single vehicle pickup and delivery problem (1-M-1 SVPDP), reviewed by Gribkovskaia and Laporte (2008). The expression "one-to-many-to-one" means that all delivery cargo is sent from the depot, and all pickup cargo is destined for the depot. This relates closely to the PSVRSP-RT in this thesis. Linehaul cargo is to be delivered at the offshore installations (destinations) whereas backhaul cargo is to be delivered at the common depot (origin) in Macaé. The 1-M-1 SVPDP is used for PSV routing in Gribkovskaia et al. (2008).

The PDPTW discussed in the articles by Christiansen clearly has certain similarities to our problem. However, it does not consider one common depot like in our case. For this reason we believe the 1-M-1 SVPDP problem discussed in Gribkovskaia and Laporte (2008) has most resemblance to the PSVRSP-RT.

An extension of the VRP that has yet to receive much research attention is known as the vehicle routing problem with intermediate replenishment facilities (VRPIRF). The problem is discussed in the articles by Crevier et al. (2007) and Tarantilis et al. (2008). The idea of the VRPIRF is that vessels are able to replenish their capacity at intermediate depots in order to extend their routes. This aspect has certain similarities to the problem described in this thesis. The difference between the VRPIRF model and the PSVRSP-RT lies in the role and characteristics of the intermediate facilities. Our model aims to properly allocate a set of tankers in order to minimize total sailing distance for each vessel, while the facilities are fixed in the VRPIRF. Additionally, the VRPIRF uses the term replenishment facility while the PSVRSP-RT focuses on refueling tankers. Thus, the intention of the refueling tankers discussed in this thesis is not to assist vessels by extending the routes, but simply a requirement that forces each vessel to visit one refueling tanker each trip. However, the article by Tarantilis et al. (2008) present an interesting extension that relates to the PSVRSP-RT.

Although there may be many ways of describing the PSVRSP-RT by naming conventions used in the literature, we believe a vehicle routing problem with pickup and delivery, soft time windows and intermediate refueling tankers (VRPPDSTWIRT) gives a proper explanation of the problem. Naturally, not all aspects of the problem are accounted for by this description.

3.2 Platform supply vessel models

Halvorsen-Weare et al. (2012) solve a fleet composition and routing problem for platform supply vessels for the energy company Statoil on the Norwegian Continental Shelf. The problem consists of identifying the optimal fleet composition of supply vessels that are to service a given number of offshore installations from one common onshore depot while at the same time determining the weekly routes and schedules for these vessels. Halvorsen-Weare et al. (2012) claim that deck capacity is the binding capacity resource on vessels, therefore all demands are given in square meters of deck space.

Halvorsen-Weare and Fagerholt (2011) create robust supply vessel schedules based on the model described in the Halvorsen-Weare et al. (2012). They incorporate weather as a stochastic factor in the periodic routing problem. A similar approach is used in a master's thesis by Aneichyk (2009). This model is also a fleet size and mix problem for the routing and scheduling of PSVs for Statoil, modelling weather as a stochastic factor.

Gribkovskaia et al. (2008) utilize the 1-M-1 SVPDP in a PSV routing problem. This article also focuses on the Norwegian Continental Shelf. The exact model description in the article is a single vehicle pickup and delivery problem with capacitated customers, originally presented in Aas et al. (2007). The only extension to the 1-M-1 SVPDP is that the offshore installations have capacity constraints. As in Halvorsen-Weare et al. (2012), the binding constraint concerning capacity on vessels is the deck area in square meters.

It is our impression after conducting a thorough literature search that there is limited available research on PSV routing and scheduling in the Brazilian oil industry. In light of the recent pre-salt discoveries and Petrobras expansion plans, Igreja et al. (2012) study the possibility of a large scale optimization model on PSV routing and scheduling to pre-salt areas using ports that are already fully utilized. The example case they describe has many similarities to the PSV routing and scheduling problem presented for Campos basin. The objective is to deliver cargo such as food, equipment and assembly parts to and from platforms by minimizing fuel consumption, size of the waiting queues in port, number of ships needed, total makespan and docking costs.

3.3 Solution methods

In the previous sections we have mentioned a few problems that resemble the problem of this thesis. The aim now is to briefly give an overview of research regarding solution approaches for solving extensions of the VRPTW and PDPTW.

The presentation of methods is threefold. Firstly, we provide references to literature related to the most extensively used methods. In particular the reviewed literature is related to dynamic programming, Dantzig-Wolfe decomposition (DWD) and column generation. Secondly, combined solution methods are presented in relation to more complex routing problems. Thirdly, we look at some alternative heuristic methods used to solve previously mentioned routing problems that resemble the PSVRSP-RT.

3.3.1 Common methods

Dynamic programming was the first optimization method applied on pickup and delivery problems (Mitrovic-Minic, 1998). Kolen et al. (1987) present some early research on this method in relation to the VRPTW. In more modern literature dynamic programming is often used to solve subproblems of more advanced vehicle routing problems.

Decomposition methods are commonly used in solving various VRPTW problems. An early study by Appelgren (1969) proved good results by applying column generation based on the Dantzig-Wolfe decomposition method. The algorithm solves a ship scheduling problem where the subproblems are solved by dynamic programming. However, the algorithm used does not guarantee integer solutions. Desrochers and Desrosiers (1992) present an exact method for solving large VRPTW problems using column generation. The algorithm manages to solve 100-customer problem instances.

Furthermore, the DWD approach is also presented in Christiansen (1999) where it is used to solve the IPDTW. The problem is decomposed into subproblems for each harbour and each ship. The subproblems are solved as shortest path problems by dynamic programming algorithms. Furthermore, integer requirements are achieved by branching on the underlying structure. Variable splitting is conducted in order to provide a better structure for decomposition. Variable splitting may also be found in the article by Fisher et al. (1997), among others. Fisher et al. (1997) describe two optimization methods for solving VRPTWs; a Langrangian decomposition method and a K-Tree relaxation method with time windows added as side constraints (Fisher, 1994). Fisher et al. (1997) conclude that the methods cannot solve as many or as varied problems as the column generation approach, but they show good performance on clustered problems. Both algorithms are able to solve problems with up to 100 customers.

3.3.2 Combined methods

There are extensive amounts of literature that focus on advanced solution approaches for solving routing problems. Many of these studies combine a set of methods in order to improve the performance of the model. The solution approaches clearly become more advanced as the models are extended beyond the basic formulation of a standard VRPTW. For instance, Andersson et al. (2011) propose three alternative solution methods to a maritime pickup and delivery problem with time windows and coupled and synchronized cargoes (PDPTW-CSC). The methods are based on a path-flow formulation and a priori column generation. Furthermore these methods are combined with a scheme for relaxing the complicating synchronization constraints and reintroducing them dynamically when needed.

Kallehauge et al. (2001) go even further than Fisher et al. (1997) in terms of problem size. A so-called cutting plane algorithm with trust-region is combined with a DW algorithm for finding integer solutions with branch and bound. The method successfully solves a VRPTW instance with 1,000 customers to optimality. Lagrangean relaxation is also a considerable part of the method used in this article and causes each subproblem to be solved as elementary shortest path problems with time windows and capacity constraints (ESPPTWCC). The results reveal that the cutting plane algorithm reduces the CPU time significantly in the root node compared to the DWD method due to easier subproblems.

Stålhane et al. (2012) introduce a branch-price-and-cut method to solve a maritime pickup and delivery problem with time windows and split loads (PDPTWSL). As for similar problems they first introduce a path-flow formulation of the problem. Dynamic programming is used to solve the subproblems and new dominance criterion is included in order to dominate partial paths. The master problem is strengthened by the use of previously known and new valid inequalities. Computational results proved that the model performs better in terms of CPU time than existing algorithms given aspects such as longer planning horizon, larger average cargo and narrower time windows.

3.3.3 Heuristics

In addition to the exact methods for providing the optimal solution, extensive research has been conducted within the area of heuristics and metaheuristics for solving advanced vehicle routing problems. Some of the most related problems reviewed in this chapter are in fact solved by the use of heuristics, and we list a few examples here.

One of these problems is the 1-M-1 SVPDP introduced in Gribkovskaia and Laporte (2008) and Gribkovskaia et al. (2008). The 1-M-1 SVPDP is solved by a so-called Unified Tabu Search (UTSA). More information regarding UTSA adaption may be found in Cordeau et al. (2001). The algorithm has proved to be one of the most successful TS algorithms for vehicle routing problems. One important feature of the UTSA search method is that solutions do not have to satisfy capacity or route length restrictions. Instead, the objective function is penalized for violations. Another article mentioned earlier is Tarantilis et al. (2008), which also makes use of heuristic solution methods. The article presents a hybrid guided local search (H-GLS) when solving the VRPIRF. The algorithm is a combination of three powerful metaheuristic strategies, i.e. variable neighborhood search (VNS), tabu search (TS) and guided local search (GLS). The computational study proved good result on benchmark VRPIRF instances used by Crevier et al. (2007).

Generally there is a clear trend towards the use of column generation within maritime ship routing problems. This chapter has shown some diversity with regard to solution methods. To learn more about other approaches to solve extensions of the PDPTW we recommend reading the paper by Mitrovic-Minic (1998). The author gives a chronological list of important publications related to dynamic programming, column generation and various forms of heuristics used for solving different types of PDPTW. In the following chapters we focus on the methods we have used for solving the PSVRSP-RT and previous literature that supports the methodology.

Chapter 4

Model Formulation

This chapter presents an arc-flow formulation of the platform supply vessel routing and scheduling problem with refueling tankers (PSVRSP-RT). The model is a continuation of the platform supply vessel routing and scheduling problem presented in Friedberg and Uglane (2012b). The only difference for this model is that we have added intermediate refueling tankers as an additional factor.

The PSVRSP-RT is formulated as a deterministic cost minimization problem. Firstly, we begin by laying out some key assumptions used in the mathematical formulation. Secondly, various symbols in the mathematical model are presented briefly. The objective function and main constraints are presented thereafter. Finally, complementary constraints, such as linearizations and valid inequalities are presented. Regarding notations, lower-case letters represent subscripts and decision variables, and capital letters represent parameters and sets.

4.1 Main assumptions

The problem description in Chapter 2 gave a general description of the problem. To make a model that resembles reality, certain assumptions are necessary. For instance, weather is known to be a decisive factor for route planning, and will for instance influence loading time and sailing speed of a vessel. The various platforms also have different capabilities with respect to rough weather. However, for the PSVRSP-RT we have assumed a deterministic planning environment. The following paragraphs explain how key elements are regarded in the model.

Costs

Cost elements are for the most part included in the objective function. The model minimizes costs, although other factors could have been considered. For instance, time and sailing distance are other possible minimization factors. The cost elements are designed to include both variable and fixed sailing costs. Furthermore, assumptions are made to account for delay and express delivery costs. These factors are included as penalties in the objective function. However, the real life implications of express costs are not known, and they are therefore not explicitly accounted for. Thus, to include the express delivery aspect a relatively high cost is assumed. This is primarily to allocate orders on regular departures.

Orders

The orders are considered known for the entire planning horizon. In reality, orders would be less predictable and generated dynamically. Still, the model formulation could mimic such an environment. By fixing the orders only for the first departure and re-running the model sequentially during the planning horizon, i.e applying a rolling horizon principle, one could simulate a more dynamic environment. For more information regarding the rolling horizon principle the reader is referred to Fagerholt et al. (2010).

Order sequencing

A vessel is assumed unable to load/unload several orders at the same time. This may not be true in all cases, but a reasonable assumption overall. Furthermore, delivery orders are handled before pickup orders to avoid deck-space limitations on the vessels during service. In reality these constraints might be less restrictive. For the refueling facilities we assume no sequencing restrictions. In practice each refueling tanker may only service two vessels at the same time.

Vessels

The PSV fleet is assumed to be fixed. The idea of a fixed fleet seems reasonable given long-term contracts with external companies. The limitations on time and distance traveled on each voyage is a way of simplifying fuel consumption for a given vessel. These parameters assume constant sailing speed. Based on size and fuel storage capacity, these limitations are difficult to measure. Weather conditions could also be decisive in this matter. To improve accuracy, it would be appropriate to include more precise parameters based on fuel consumption relative to speed. However, for the PSVRSP-RT in this thesis we have neglected this factor as we have assumed fixed speed.

Time windows

Every order is given an inner time window for delivery or pickup. Service cannot start earlier than the inner time window, thus the lower time window is fixed. The upper time window is soft to allow for delayed deliveries. Delivering an order overdue leads to a penalty, or inconvenience, which is proportional to the amount of delay. Fagerholt (2001) describes a few penalty functions used in related problems. The author mentions linear, fixed and quadratic penalty functions. For our model formulation we have assumed a linear penalty function, which is shown in Figure 4.1. An upper fixed time window also exists. This states the maximum delay allowed. If the model violates an upper time window, an express delivery of that order is initiated. Since the formulation does not explicitly model express delivery, it is considered a last resort for all orders. Furthermore, the model does not include opening hours for individual platforms although certain facilities may be closed down at certain hours during the day, e.g. at night.

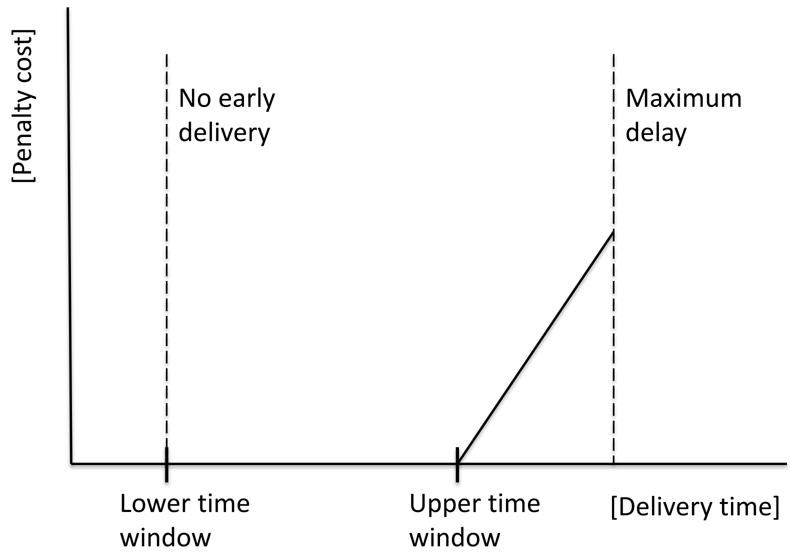

Figure 4.1: *Penalty function for the PSVRSP-RT*

Capacity

The deck space area is assumed to be the binding capacity constraint for a vessel, because most cargo is loaded on one level. Capacity restrictions such as dead weight tons or volume restrictions are therefore neglected in the PSVRSP-RT. The model does not account for bulk cargo such as cement, fuel and water, similar to the models presented in Halvorsen-Weare et al. (2012) and Friedberg and Uglane (2012b).

4.2 Mathematical formulation - Arc-flow model

In this section we present the notations and the arc-flow formulation of the PSVRSP-RT. The nodes represent offshore facilities and refueling buoys. The term arc is used to describe a sailing path between two nodes or a node and the depot. The common depot in this case, refers to the port in Macaé.

Sets and indices

$i \in \mathcal{N}$	Set of all nodes, including the depot $\{0\}$
$i \in \mathcal{N}^P \subseteq N$	Set of offshore installations
$i \in \mathcal{N}^B \subseteq N$	Set of refueling buoys
$(i,j) \in \mathcal{A}$	Set of arcs
$o \in \mathcal{O}_i$	Set of orders for node i
$o \in \mathcal{O}_i^P$	Set of pickup orders for node i
$o \in \mathcal{O}_i^D$	Set of delivery orders for node i
$v \in \mathcal{V}$	Set of vessel types
$d \in \mathcal{D}$	Set of departures

Parameters

A_o	Area required (pickup and delivery) for order o
B_{ij}	Distance between node i and j
C_v^F	Fixed cost for vessel type v
C_o^P	Penalty per time overdue for order o
C^X	Express delivery penalty
Q_v	Number of vessels for vessel type v available in the planning period
K_v	Capacity (m^2) for vessel type v
L_v^T	Limit on sailing time for vessel type v
L_v^D	Limit on sailing distance for vessel type v
$\overline{T}_o$	Latest start of service for order o without penalty
$\underline{T}_o$	Earliest start of service for order o
T_o^S	Service time at platform upon visit for order o
T_d^D	Departure time for departure d
T^M	Maximum delay allowed
C_{ijv}^V	Sailing cost between node i and j by vessel type v
T_{ijv}^V	Sailing time between node i and j by vessel type v
T^F	Time required for refueling operations
G	Available fleet of refueling tankers

Variables

x_{ijd}	1 if a vessel travels from i to j on departure d, 0 otherwise
y_{od}	1 if order o is delivered on departure d, 0 otherwise
w_o	1 if express delivery is needed for order o, 0 otherwise
t_{id}	Departure time from node i for departure d
s_o	Start time of service for order o
d_o	Arrival delay for order o
l_{id}	Outbound load at node i on departure d
c_d	Variable sailing cost for departure d
v_{vd}	1 if vessel type v is chosen for departure d, 0 otherwise
h_{op}	1 if order o is serviced before order p, 0 otherwise
e_i	1 if a tanker is deployed at node i, 0 otherwise
f_{id}	1 if departure d refuels at node i

Objective function

$$\min z = \sum_{d \in \mathcal{D}} c_d + \sum_{v \in \mathcal{V}} \sum_{d \in \mathcal{D}} C^F_v v_{vd} + \sum_{i \in \mathcal{N}^P} \sum_{o \in \mathcal{O}_i} C^P_o d_o + \sum_{i \in \mathcal{N}^P} \sum_{o \in \mathcal{O}_i} C^X w_o \tag{4.1}$$

The objective function (4.1) minimizes total sailing costs and penalties for all departures. The first two terms represent the variable and fixed sailing costs respectively. The last two terms ensure that delayed orders and express deliveries are penalized.

Variable sailing cost constraints

$$c_d - \sum_{(i,j) \in \mathcal{A}} C^V_{ijv} x_{ijd} v_{vd} \geq 0 \qquad v \in \mathcal{V}, d \in \mathcal{D} \tag{4.2}$$

Constraints (4.2) represent a lower bound on the total sailing cost for a given departure in the objective function. The intention of this representation is to avoid nonlinear terms in the objective function. Constraints (4.43) present the linearizations of these constraints.

Flow network constraints

$$\sum_{j \in \mathcal{N}^P} x_{ijd} \leq 1 \qquad i \in \mathcal{N}, d \in \mathcal{D} \tag{4.3}$$

$$\sum_{i \in \mathcal{N}} x_{ijd} = f_{jd} \qquad j \in \mathcal{N}^B, d \in \mathcal{D} \tag{4.4}$$

$$f_{id} \leq e_i \qquad i \in \mathcal{N}^B, d \in \mathcal{D} \tag{4.5}$$

$$\sum_{i \in \mathcal{N}^B} f_{id} = 1 \qquad d \in \mathcal{D} \tag{4.6}$$

$$\sum_{i \in \mathcal{N}} x_{ijd} - \sum_{i \in \mathcal{N}} x_{jid} = 0 \qquad j \in \mathcal{N}, d \in \mathcal{D} \tag{4.7}$$

$$\sum_{j \in \mathcal{N}} x_{0jd} = 1 \qquad d \in \mathcal{D} \tag{4.8}$$

$$\sum_{i \in \mathcal{N}} x_{i0d} = 1 \qquad d \in \mathcal{D} \tag{4.9}$$

Constraints (4.3) ensure that a vessel leaving a node travels to no more than one node in the set $\mathcal{N}^P$ for any departure. In constraints (4.4) we make sure that a vessel only travels to a refueling buoy if the vessel refuels there. Constraints (4.5) say that a vessel can only refuel at a buoy if there is a tanker deployed there. Constraints (4.6) ensure that at least one buoy is visited on each departure. In constraints (4.7) the flow conservation constraints make sure that every node with an incoming arc must have an outgoing arc. Finally, constraints (4.8) and (4.9) respectively ensure that a vessel begins and ends at the depot.

Fleet choice constraints

$$\sum_{v \in \mathcal{V}} v_{vd} = 1 \qquad d \in \mathcal{D} \tag{4.10}$$

$$\sum_{d \in \mathcal{D}} v_{vd} \leq Q_v \qquad v \in \mathcal{V} \tag{4.11}$$

$$\sum_{i \in \mathcal{N}^B} e_i \leq G \tag{4.12}$$

Constraints (4.10) make sure that only one vessel type is picked for any departure. Furthermore, constraints (4.11) limit the number of vessels of each type used in the planning period. Lastly, constraints (4.12) ensure that the number of tankers deployed is less than or equal to the available fleet of tankers.

Order balance constraints

$$y_{od} - \sum_{i \in \mathcal{N}} x_{ijd} \leq 0 \qquad j \in \mathcal{N}^P, o \in \mathcal{O}_j, d \in \mathcal{D} \tag{4.13}$$

$$\sum_{o \in \mathcal{O}_j} y_{od} - \sum_{i \in \mathcal{N}} x_{ijd} \geq 0 \qquad j \in \mathcal{N}^P, d \in \mathcal{D} \tag{4.14}$$

$$\sum_{d \in \mathcal{D}} y_{od} + w_o = 1 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{4.15}$$

Constraints (4.13)-(4.15) connect the arc flow variables with the order delivery variables. Constraints (4.13) ensure that an order can only be delivered if the respective node is visited. Constraints (4.14) make sure that a vessel only may travel to a node j if at least one order is delivered, and constraints (4.15) ensure that all orders are fulfilled. The express variable, w_o, in (4.15) is equal to 1 for orders not included in the ordinary departures.

Order sequencing constraints

$$h_{op} + h_{po} - y_{od} - y_{pd} \geq -1 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i, d \in \mathcal{D} \tag{4.16}$$

$$h_{op} + h_{po} \leq 1 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i \tag{4.17}$$

$$h_{op} - h_{po} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i^D, p \in \mathcal{O}_i^P \tag{4.18}$$

Nodes containing several orders are assigned service sequences. Constraints (4.16) ensure that at least one sequence is chosen if both orders (o and p) are delivered at node i on departure d. The constraints (4.17) assure that no more than one sequence is picked for a given order pair. Furthermore, constraints (4.18) separate pickup and delivery orders by implying that delivery should be conducted before pickup orders.

Time constraints

$$\underline{T}_o \leq s_o \leq \overline{T}_o + T^M \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{4.19}$$

$$s_p - (s_o + T_o^S)\, h_{op} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i \tag{4.20}$$

$$t_{0d} = T_d^D \qquad d \in \mathcal{D} \tag{4.21}$$

$$t_{id} - (s_o + T_o^S)\, y_{od} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, d \in \mathcal{D} \tag{4.22}$$

$$s_o - (t_{id} + \sum_{v \in \mathcal{V}} T_{ijv}^V v_{vd})\, x_{ijd}\, y_{od} \geq 0 \qquad i \in \mathcal{N}, j \in \mathcal{N}^P, o \in \mathcal{O}_j, d \in \mathcal{D} \tag{4.23}$$

$$t_{jd} - (T^F + t_{id} + \sum_{v \in \mathcal{V}} T_{ijv}^V v_{vd}) x_{ijd} \geq 0 \qquad i \in \mathcal{N}^P \cap \{0\}, j \in \mathcal{N}^B, d \in \mathcal{D} \tag{4.24}$$

Constraints (4.19) give the time window for start of service for all orders. The lower bound is hard, while the maximum time delay parameter, T^M, sets the upper bound for delay. Constraints (4.20) ensure start of service for two consecutive orders. In (4.45), the linearizations of these constraints are shown. Constraints (4.21) say that the departure time from the depot is the fixed departure time. Constraints (4.22) make sure that an order is serviced before a vessel continues a voyage. Constraints (4.23) contain an inequality sign because waiting time is permitted prior to start of service. The constraints also assure that start of service at any node is restricted by the departure and sailing time from the previous node. Lastly, they ensure subtour elimination. The constraints are linearized in (4.49). Constraints (4.24) ensure that the departure time at refueling buoys is restricted by the time of refueling operations, and the departure time and sailing time from the previous node. The linearization of these these constraints are given in (4.51).

Limit on sailing time

$$t_{id} + \sum_{v \in \mathcal{V}} (T^V_{i0v} - L^T_v)\, v_{vd} \leq T^D_d \qquad i \in \mathcal{N}, d \in \mathcal{D} \tag{4.25}$$

Constraints (4.25) ensure that any vessel arrives at the destination (depot) node within the time limit restricted by the vessel in use.

Time delay constraints

$$s_o - d_o \leq \overline{T}_o \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{4.26}$$

Constraints (4.26) put a lower bound on the arrival delay variable d_o.

Capacity constraints

$$l_{id} - \sum_{v \in \mathcal{V}} K_v\, v_{vd} \leq 0 \quad i \in \mathcal{N}, d \in \mathcal{D} \tag{4.27}$$

$$\sum_{i \in \mathcal{N}^P} \sum_{o \in \mathcal{O}^D_i} A_o\, y_{od} - l_{0d} = 0 \quad d \in \mathcal{D} \tag{4.28}$$

$$x_{ijd}\, (l_{id} + \sum_{o \in \mathcal{O}^P_j | i \in \mathcal{N}^P} A_o\, y_{od} - \sum_{o \in \mathcal{O}^D_j | i \in \mathcal{N}^P} A_o\, y_{od} - l_{jd}) = 0 \quad (i,j) \in \mathcal{A}, d \in \mathcal{D} \tag{4.29}$$

Constraints (4.27) assure that outbound load at a given node does not exceed the vessel capacity. Constraints (4.28) set the outbound load from the depot equal to the sum of all outgoing delivery orders. The nonlinear load balance constraints are given by (4.29). Constraints (4.29) are linearized by (4.51) and (4.53).

Limit on sailing distance

$$\sum_{(i,j)\in\mathcal{A}} B_{ij}\, x_{ijd} - \sum_{v\in\mathcal{V}} L_v^D v_{vd} \leq 0 \qquad d \in \mathcal{D} \tag{4.30}$$

Constraints (4.30) are sailing distance constraints given for each departure. Each vessel type has an upper limit for maximum sailing distance and (4.30) ensure that each route holds this restriction.

Variable constraints

$$x_{ijd} \in \{0,1\} \qquad (i,j) \in \mathcal{A}, d \in \mathcal{D} \tag{4.31}$$
$$y_{od} \in \{0,1\} \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, d \in \mathcal{D} \tag{4.32}$$
$$w_o \in \{0,1\} \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{4.33}$$
$$t_{id} \geq 0 \qquad i \in \mathcal{N}, d \in \mathcal{D} \tag{4.34}$$
$$s_o \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{4.35}$$
$$d_o \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{4.36}$$
$$l_{id} \geq 0 \qquad i \in \mathcal{N}, d \in \mathcal{D} \tag{4.37}$$
$$c_d \geq 0 \qquad d \in \mathcal{D} \tag{4.38}$$
$$v_{vd} \in \{0,1\} \qquad v \in \mathcal{V}, d \in \mathcal{D} \tag{4.39}$$
$$h_{op} \in \{0,1\} \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i \tag{4.40}$$
$$e_i \in \{0,1\} \qquad i \in \mathcal{N}^B \tag{4.41}$$
$$f_{id} \in \{0,1\} \qquad i \in \mathcal{N}^B, d \in \mathcal{D} \tag{4.42}$$

The arc flow, delivery and express variables are binary and respectively illustrated in (4.31)-(4.33). Constraints (4.34)-(4.38) ensure non-negativity for time, load and cost variables. Vessel choice and sequencing variables are binary and given by (4.39) and (4.40). The tanker deployment variables and refueling variables are also binary and represented by constraints (4.41) and (4.42) respectively.

4.2.1 Linearizations

The model presented so far includes nonlinearities in a number of constraints. In this subsection the linearizations of the respective constraints are laid out.

Variable sailing cost constraints (4.2)

$$c_d - \sum_{(i,j)\in\mathcal{A}} C_{ijv}^V x_{ijd} - M_v^1 v_{vd} \geq -M_v^1 \qquad v \in \mathcal{V}, d \in \mathcal{D} \tag{4.43}$$

$$M_v^1 = C_v^S L_v^D \qquad v \in \mathcal{V} \tag{4.44}$$

The linearization of (4.2) is ensured by (4.43). The constraints is redundant whenever v_{vd} is zero. M_v^1 is given by (4.44), which represent the highest sailing cost possible for a vessel.

Time constraints (4.20)

$$s_p - s_o - (M_{op}^2 + T_o^S)h_{op} \geq -M_{op}^2 \qquad i \in \mathcal{N}^P o \in \mathcal{O}_i, p \in \mathcal{O}_i \qquad (4.45)$$

$$\begin{aligned} M_{op}^2 &= \max\{s_o - s_p\} \\ &= \overline{T}_o + T^M - \underline{T}_p \end{aligned} \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i \qquad (4.46)$$

Constraints (4.45) are redundant whenever the sequencing variable is zero. Constraints (4.46) ensure redundancy by fixing M_{op}^2 in a manner that allows order p to be serviced earlier if no sequencing is required.

Time constraints (4.22)

$$t_{id} - s_o - M_{op}^3 y_{od} \geq T_o^S - M_{op}^3 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, d \in \mathcal{D} \qquad (4.47)$$

$$\begin{aligned} M_{opd}^3 &= \max\{s_o + T_o^S - t_{id}\} \\ &= \overline{T}_o + T^M + T_o^S - T_d^D - \min_{v \in \mathcal{V}}\{T_{0iv}^V\} \end{aligned} \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, d \in \mathcal{D} \qquad (4.48)$$

Constraints (4.47) are redundant whenever the delivery variable is zero. Constraints (4.48) ensure redundancy by fixing M_{op}^3 as the last possible departure time for a node.

Time constraints (4.23)

$$\begin{aligned} & s_o - t_{id} - \sum_{v \in \mathcal{V}} T_{ijv}^V v_{vd} \\ & - M_{ijod}^4 (x_{ijd} + y_{od}) \geq -2M_{ijod}^4 \end{aligned} \qquad i \in \mathcal{N}, j \in \mathcal{N}^P, o \in \mathcal{O}_j, d \in \mathcal{D} \qquad (4.49)$$

$$\begin{aligned} & M_{ijod}^4 = T_d^D - \underline{T_o} \\ & + \max_{v \in \mathcal{V}}\{L_v^T - T_{i0v}^V + T_{ijv}^V\} \end{aligned} \qquad i \in \mathcal{N}, j \in \mathcal{N}^P, o \in \mathcal{O}_j, d \in \mathcal{D} \qquad (4.50)$$

Constraints (4.49) have a more complicated representation due to the number of variables. The constraints are redundant whenever the arc flow and delivery variables are zero. In other words, whenever a node is not visited, thus no delivery, start of service is unrestricted. Constraints (4.50) put M_{ijod}^4 equal to the latest possible start of service for a given node.

Time constraints (4.24)

$$t_{jd} - t_{id} - \sum_{v \in \mathcal{V}} T^V_{ijv} v_{vd} - M^5_{ijd} x_{ijd} \geq T^F - M^5_{ijd} \qquad i \in \mathcal{N}^P \cap \{0\}, j \in \mathcal{N}^B, d \in \mathcal{D} \tag{4.51}$$

$$M^5_{ijd} = T^D_d + T^F + \max_{v \in \mathcal{V}} \{L^T_v - T^V_{i0v} + T^V_{ijv}\} \qquad i \in \mathcal{N}^P \cap \{0\}, j \in \mathcal{N}^B, d \in \mathcal{D} \tag{4.52}$$

Constraints (4.51) are redundant whenever a vessel does not travel between a platform and a refueling tanker, i.e. the arc flow variables are zero. The constraints (4.52) ensure redundancy by enforcing M^5_{ijd} to be equal to latest possible departure time at a given platform node i.

Capacity constraints (4.29)

$$l_{id} + \sum_{o \in \mathcal{O}^P_j | j \in \mathcal{N}^P} A_o y_{od} - \sum_{o \in \mathcal{O}^D_j j \in \mathcal{N}^P} A_o y_{od} - l_{jd} + M^6 x_{ijd} \geq M^6 \qquad (i,j) \in \mathcal{A}, o \in \mathcal{O}_i, d \in \mathcal{D} \tag{4.53}$$

$$l_{id} + \sum_{o \in \mathcal{O}^P_j j \in \mathcal{N}^P} A_o y_{od} - \sum_{o \in \mathcal{O}^D_j j \in \mathcal{N}^P} A_o y_{od} - l_{jd} + M^6 x_{ijd} \leq M^6 \qquad (i,j) \in \mathcal{A}, o \in \mathcal{O}_i, d \in \mathcal{D} \tag{4.54}$$

$$M^6 = \max_{v \in \mathcal{V}} \{K_v\} \tag{4.55}$$

The linearization of the load balance constraints are given by (4.53) and (4.54). Redundancy is assured by (4.55), where M^6 is fixed as the vessel load capacity.

4.2.2 Valid inequalities

In this section we introduce valid inequalities that may contribute to a tighter formulation of the model. A tighter formulation might lead to an improved solution process by decreasing the CPU time.

$$c_d - \sum_{(i,j) \in \mathcal{A}} \min_{v \in \mathcal{V}} \{C^V_{ijv}\} x_{ijd} \geq 0 \qquad d \in \mathcal{D} \tag{4.56}$$

Constraints (4.56) ensure that the total variable cost is at minimum equal to the cost of using the cheapest vessel type. These constraints may be included to contribute to a stronger representation of the variable sailing cost constraints (4.43).

$$\sum_{i\in\mathcal{S}}\sum_{j\in\mathcal{S}} x_{ijd} \leq |\mathcal{S}| - 1 \qquad \mathcal{S} \subseteq \mathcal{N}\backslash\{0\}, |\mathcal{S}| \geq 2, |\mathcal{S} \cap \mathcal{N}^B| \leq 1, d \in \mathcal{D} \tag{4.57}$$

Constraints (4.57) are subtour elimination constraints of the VRP (Toth and Vigo, 2002). They ensure that for each departure d, a cycle cannot be created in a subset of nodes $\mathcal{S}$ that does not contain the depot. Furthermore, a subset may only contain maximum one refueling buoy.

$$\sum_{i\in\overline{\mathcal{S}}}\sum_{j\in\mathcal{S}} x_{ijd} - y_{od} \geq 0 \qquad \mathcal{S} \subseteq \mathcal{N}\backslash\{0\}, |\mathcal{S}| \geq 2, k \in \mathcal{S}, o \in \mathcal{O}_k, d \in \mathcal{D} \tag{4.58}$$

Constraints (4.58) are another version of subtour elimination constraints, modified for this particular problem. They state that the sum of arcs entering a subset $\mathcal{S}$ where an order o of that set is delivered, must be greater than or equal to 1.

$$t_{id} - \sum_{v\in\mathcal{V}} T^V_{0iv} v_{vd} \geq T^D_d + \min_{o\in\mathcal{O}_i}\{T^S_o\} \qquad i \in \mathcal{N}^P, d \in \mathcal{D} \tag{4.59}$$

$$t_{id} - \sum_{v\in\mathcal{V}} T^V_{0iv} v_{vd} \geq T^D_d + T^F \qquad i \in \mathcal{N}^B, d \in \mathcal{D} \tag{4.60}$$

$$s_o - (T^D_d + \min_{v\in\mathcal{V}}\{T^V_{0iv}\}) y_{od} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, d \in \mathcal{D} \tag{4.61}$$

$$t_{id} - (T^D_d + T^F + \min_{v\in\mathcal{V}}\{T^V_{0iv}\}) f_{id} \geq 0 \qquad i \in \mathcal{N}^B, d \in \mathcal{D} \tag{4.62}$$

Constraints (4.59) say that the departure time at a platform node i must be greater than or equal to the first possible visit to node i on departure d in addition to the minimum of all service times for that node. The same principle is used in constraints (4.60), but in this case we look at nodes associated with refueling buoys. Constraints (4.61) make sure that the start of service for order o at platform node i is at least the time of the first possible visit to node i given that the order is delivered on departure d. Lastly, constraints (4.62) involve buoy nodes and put a lower bound on the departure times given that a tanker is deployed at the buoy.

Chapter 5

Dantzig-Wolfe Decomposition

In this chapter we introduce the Dantzig-Wolfe decomposition (DWD) method and how it is applied to the PSVRSP-RT. Firstly, in Section 5.1 we go through some of the basic principles of DWD and briefly mention other decomposition methods. Secondly, in Section 5.2 the link between the DW applications is discussed in relation to solving mixed integer linear programs (MILP). Finally, in Section 5.3 the methodology is used to reformulate the arc-flow model of Chapter 4 based on DWD.

5.1 Basic principles of DWD

Decomposition methods was developed as a way of enhancing the solution process when solving large models. The principle of decomposition refers to the idea of solving a sequence of smaller and easier problems instead of one large problem (Lundgren et al., 2010).

Certain problems are more suitable with regard to structure and therefore easier to decompose. In this case, structure typically means the pattern of zero an nonzero coefficients in the constraints. Bradley et al. (1977) explain some of the most important patterns that frequently appear in various problems. The patterns are illustrated in Figure 5.1. For more information about different decomposition strategies in large-scale problems, we suggest reading Bradley et al. (1977) and Elster (1993).

Based on the problem structure there may be alternative decomposition methods. Arguably, the three most common methods are Dantzig-Wolfe decomposition, Lagrangean relaxation (LR) and Benders decomposition (BD) (Gunnerud, 2011). In this thesis we have only considered DWD, as it has proven good results in previous research related to maritime transportation problems (see Chapter 3). However, the other two methods could also be applied to this problem. LR is in fact a contributing factor for solving a VRPTW in Kallehauge et al. (2001). The equivalence between between DWD and LR is well known and is described in more detail by Letocart et al. (2012) and Lemaréchal (2007). Both LR and DWD are suited for problems with a block angular constraint structure (Gunnerud et al., 2009). The structure is illustrated in Figure 5.1 c). The principles of LR and BD are explained in some detail in Beasley (1993) and Benders (1962), respectively.

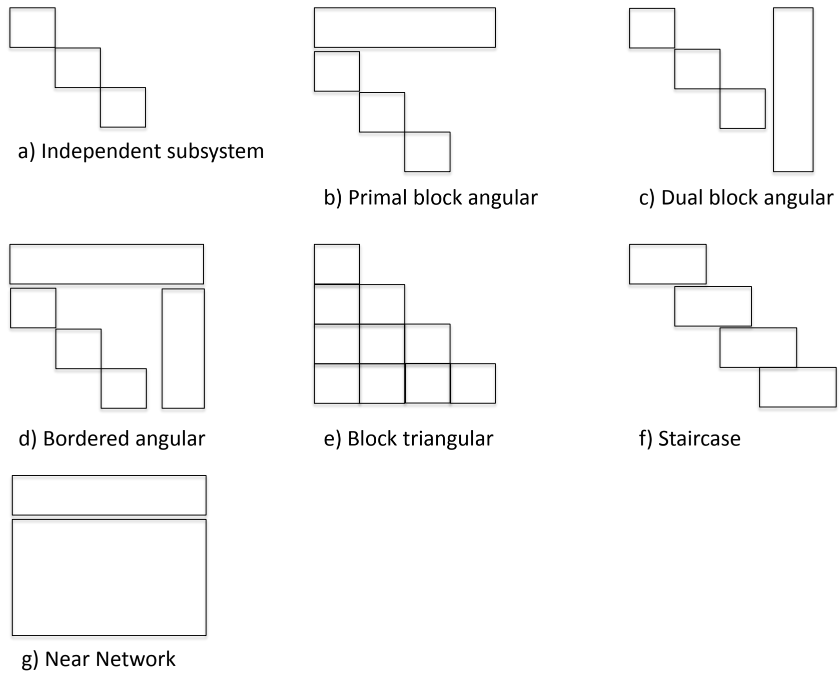

Figure 5.1: *Illustration of frequently used constraint patterns in optimization problems (Bradley et al., 1977)*

The Dantzig-Wolfe decomposition method was first developed and published by George B. Dantzig and Phillip Wolfe (Dantzig and Wolfe, 1960). The set of feasible solutions to a problem may be described in two ways. Firstly, an outer representation means that the feasible region is described as the intersection of halfspaces. Secondly, a solution could have an interior representation, in which case the feasible region is described as a convex combination of extreme points. The original model in Section 4.2 uses an outer representation, while DWD requires an interior representation.

The fundamental idea of DWD is best shown through an example of the reformulation and solution process. We illustrate the principle by looking at a simplified problem (P). given by (5.1)-(5.4). In the original formulation (P) constraints (5.3) have a simple structure and are easy to solve, while constraints (5.2) are complex and destroy the simple structure. The formulation (P) is an outer representation as the feasible region is bounded by constraints (5.2) and (5.3) .

$$(P) \ \min z = c^T x \tag{5.1}$$

$$s.t. \ Ax \geq b \tag{5.2}$$

$$Dx \geq e \tag{5.3}$$

$$x \geq 0 \tag{5.4}$$

Given the simple structure of constraints (5.3), we have that the problem only consisting of these constraints are easily solvable and can be written as

$$\min\ c^T x \tag{5.5}$$
$$s.t.\ Dx \geq e \tag{5.6}$$
$$x \geq 0 \tag{5.7}$$

Furthermore, we have that $X_D = \{x|Dx \geq e, x \geq 0\}$ is bounded and the extreme points to the set X_D are denoted $x^{(1)}$, $x^{(2)}$,...,$x^{(p)}$, where p is typically a large number. A point x ($x \in X_D$) can be expressed as

$$x = \sum_{j=1}^{p} \lambda_j x^{(j)} \tag{5.8}$$
$$\sum_{j=1}^{p} \lambda_j = 1 \tag{5.9}$$
$$\lambda_j \geq 0 \qquad j = 1, \ldots, p \tag{5.10}$$

The formulation above is an interior representation and our original problem (P) can now be rewritten using the extreme points. The representation given by (5.11)-(5.14) is known as the master problem (MP).

$$(MP)\ \min z = \sum_{j=1}^{p} (c^T x^{(j)})\lambda_j \tag{5.11}$$
$$s.t.\ \sum_{j=1}^{p} (Ax^{(j)})\lambda_j \geq b \tag{5.12}$$
$$\sum_{j=1}^{p} \lambda_j = 1 \tag{5.13}$$
$$\lambda_j \geq 0, j = 1, ..., p \tag{5.14}$$

Notice that $c^T x^{(j)}$ and $Ax^{(j)}$ are constants, while λ_j is a variable. Furthermore, we have removed the simple constraints (5.3). Although we have less constraints, we are left with more variables. As mentioned, p is a large number and refers to the number of extreme points. We therefore consider that we have q known extreme points instead, where $q \ll p$. In a practical problem, we will often have only a few extreme points that are a part of the solution and we therefore do not need to generate them all. The formulation given by (5.15)-(5.18) is known as the restricted master problem (RMP).

$$(RMP)\ \min z_q = \sum_{j=1}^{q}(c^T x^{(j)})\lambda_j \tag{5.15}$$

$$s.t.\ \sum_{j=1}^{q}(Ax^{(j)})\lambda_j \geq b\,|v \tag{5.16}$$

$$\sum_{j=1}^{q}\lambda_j = 1\,|u \tag{5.17}$$

$$\lambda_j \geq 0, j = 1, ..., q \tag{5.18}$$

By solving the RMP and collecting a temporary optimal solution given by $\overline{\lambda}_j$, ($j = 1, \ldots, q$) with dual variables $\overline{v}$ and $\overline{u}$, we may acquire a feasible solution

$$\overline{x} = \sum_{j=1}^{q}\overline{\lambda}_j x^{(j)} \tag{5.19}$$

for the problem (P). Whether the solution $\overline{x}$ is the optimal solution to (P) relies on the solution of the subproblem (SP) given by (5.20)-(5.22). For the minimization problem, the objective is to find the smallest reduced cost $\overline{c}_{q+1}$ to the problem in order to decide whether we need to expand the RMP with another extreme point and resolve the problem.

$$(SP)\ \min \overline{c}_{q+1} = (c^T - \overline{v}^T A)x - \overline{u} \tag{5.20}$$

$$s.t.\ Dx \geq e \tag{5.21}$$

$$x \geq 0 \tag{5.22}$$

If $\overline{c}_{q+1} \geq 0$ we know that $\overline{x}$ is an optimal solution to P, and if $\overline{c}_{q+1} < 0$, we need to extend the RMP. The iteration process is described in Figure 5.2.

The DWD model takes advantage of the block angular structure of the problem. The SP will decompose in as many smaller problems as there are numbers of blocks. Figure 5.3 shows how this concept relates to problem (P). In the figure we have n blocks and therefore get n separate subproblems. The RMP is extended by one variable (column) in each iteration for every SP that generates a negative reduced cost.

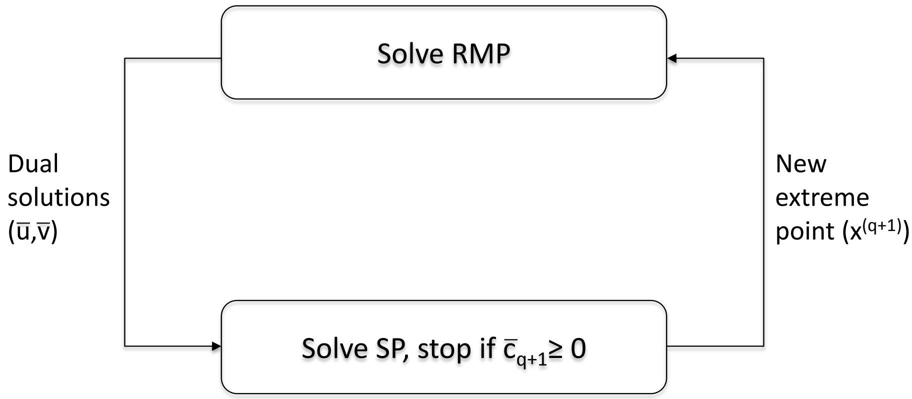

Figure 5.2: *Iteration process for the Dantzig-Wolfe method*

Algorithm 1 gives a more formal description of the solution process when solving the DWD model using column generation. In this section we have used the term extreme point in relation to a SP solution. However, in the remainder of this thesis we use the term column as the overall denomination for the SP solution. For larger problem instances a column may contain many solution variables, so the term extreme point might be somewhat understated. For future reference, the term DWD will be used to describe a decomposed model with interior representation. Furthermore, we use the terms DW algorithm and column generation interchangeably to describe the column generation scheme applied to solve the DWD model.

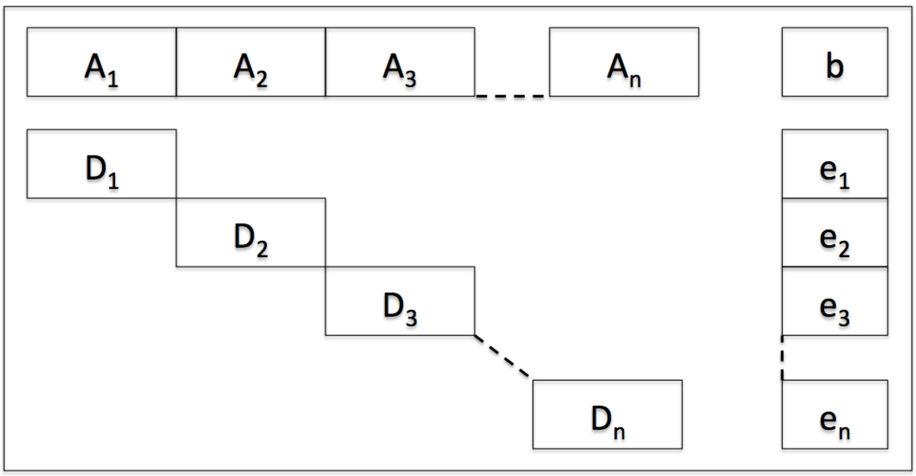

Figure 5.3: *Illustration of the block angular structure for problem (P)*

Algorithm 1: Column generation for a DWD model

Start with a subset of columns, $x^{(j)}, j = 1, ..., q$
repeat
  forall the *common constraints in RMP* **do**
    Find the primal solution λ_j and the corresponding dual solutions $\overline{v}$ and $\overline{u}$.
    The objective function value z_q provides a pessimistic bound for the optimal objective function value z^*.
  Send the dual variables, $\overline{v}$ and $\overline{u}$, to the subproblems
  forall the *subproblems* **do**
    Formulate and solve the SP.
    Provide the solution $x^{(q+1)}$ with objective function value $\overline{c}_{q+1}$
    if *a column has negative reduced cost, i.e.* $\overline{c}_{q+1} < 0$ **then**
      Add the column $x^{(q+1)}$ to RMP and introduce a new variable λ_{j+1}.
  Update the counter $q := q + 1$
until *no new columns with negative reduced cost can be found*;
Solve the RMP and provide the optimal solution z^*

5.2 Solving MILPs using DWD

In Section 5.1 we explained how Dantzig-Wolfe decomposition is used to systematically decompose a large LP problem into a master problem and smaller subproblems. However, the PSVRSP-RT presented in this thesis is a mixed integer linear problem (MILP) and we therefore have to make certain precautions when this method is applied.

An important aspect of the DW algorithm is that it only guarantees an optimal solution for the LP relaxation of an IP problem. Thus, when the problem is solved using the DW method we get a lower bound on the objective function value (minimization problem). Hence, a motivation for applying a DW algorithm to a MILP is to strengthen the solution of the LP relaxation.

Integer variables can be handled in several ways. For instance, we may apply an heuristic approach after generating a sufficient number of basic feasible points. A feasible solution can simply be found by demanding integer values for the decision variables and solving the RMP as an IP problem (Gunnerud, 2011). One well known approach to retrieve optimal integer solutions is by the use of Branch and Price (B&P) (Barnhart et al., 1998). This method is explained further in Chapter 6.

5.3 Path-flow formulation

In this section we present the reformulation of the arc-flow model from Chapter 4. A so-called path-flow formulation is used to take advantage of the underlying structure when the problem is decomposed. The first subsection presents the transformation from the arc-flow to the path-flow model. Furthermore, the decomposed model is presented based on the DWD approach.

5.3.1 Decomposition strategy

Looking at the underlying structure of the problem, we see that a majority of the constraints are defined for each departure. It therefore seems evident that each departure can be solved individually as a subproblem. In fact, the only constraint binding the set of departures are the fleet choice constraints (4.11) and the order balance constraints (4.15). The constraints are respectively illustrated below.

$$\sum_{d \in \mathcal{D}} v_{vd} \leq Q_v \qquad v \in \mathcal{V}$$

$$\sum_{d \in \mathcal{D}} y_{od} + w_o = 1 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i$$

The problem can be further decomposed by including subproblems for each vessel, which will expand our set of common constraints. Hence, the fleet choice constraint (4.10) must also be included in the master problem.

$$\sum_{v \in \mathcal{V}} v_{vd} = 1 \qquad d \in \mathcal{D}$$

Decomposition based on vessel type is in line with the method used in Christiansen (1999), while the departure time aspect is more related to the characteristics of the PSVRSP-RT. However, the departures may in fact be viewed as individual depots, although they are located at the same geographical position. Decomposition based on different depots is a common approach in ship routing problems.

In the PSVRSP-RT we also have the aspect of refueling tankers and express vessels. We therefore need to decide how to incorporate the tanker deployment variable e_i and the express delivery variable w_o in the model. The flow network constraints (4.5) and the fleet choice constraints (4.12) essentially decide the deployment of the refueling tankers (represented by e_i), and are included among the common constraints in the master problem. This approach is necessary because the tanker deployment is binding for all subproblems, i.e. all (v, d) combinations. Express delivery w_o is only represented in (4.15) and is therefore included in the master problem as well.

$$f_{id} \leq e_i \qquad i \in \mathcal{N}^B, d \in \mathcal{D}$$

$$\sum_{i \in \mathcal{N}^B} e_i \leq G$$

Besides the chosen decomposition model, the problem could alternatively be decomposed into subproblems for each node and/or order. In either case, this would increase the number of subrpoblems, while also reducing the size of each them. Furthermore, the master problem would also increase in size due to more variables. This is because the input parameters will usually include more nodes and orders than departures.

Gunnerud (2011) mentions two important aspects for choosing constraints to include in the subproblem. Firstly, they should not be overly complicated in order to assist re-optimization. Secondly, the quality of the bounds created are important during the solution process. In general, the model should provide a reasonable balance between the size of the master problem and subproblems. In most routing problems the master problem is usually much smaller and solved relatively easy. Thus, the challenge revolves around solving the subproblems efficiently. For the PSVRSP-RT, the remaining constraints not mentioned above are included in each subproblem. This decomposition strategy suggests that the subproblems are significantly larger than the corresponding master problem.

5.3.2 Variable splitting

If the arc-flow model were to be decomposed directly, it would not fall apart based on the variable indexation. To establish subproblems we need to be able to break up the corresponding constraints, which requires all subproblem variables to include an index for each departure d and vessel v. The arc-flow model states that we have a vast variety of variables, and very few of them are defined for both departures and vessels. To cope with this we introduce variable splitting (Fisher et al., 1997).

Constraints (5.23)-(5.30) show the relation between the variables from the arc-flow model and the new variables with (v, d) combinations. This method builds on the same principles as the approach used in Christiansen (1999) and Fisher et al. (1997), for a IPDPTW and VRPTW, respectively.

$$x_{ijd} - \sum_{v \in \mathcal{V}} x_{ijvd} = 0 \qquad (i,j) \in \mathcal{A}, d \in \mathcal{D} \tag{5.23}$$

$$y_{od} - \sum_{v \in \mathcal{V}} y_{ovd} = 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, d \in \mathcal{D} \tag{5.24}$$

$$t_{id} - \sum_{v \in \mathcal{V}} t_{ivd} = 0 \qquad i \in \mathcal{N}, d \in \mathcal{D} \tag{5.25}$$

$$s_o - \sum_{v \in \mathcal{V}} \sum_{d \in \mathcal{D}} s_{ovd} = 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{5.26}$$

$$d_o - \sum_{v \in \mathcal{V}} \sum_{d \in \mathcal{D}} d_{ovd} = 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{5.27}$$

$$l_{id} - \sum_{v \in \mathcal{V}} l_{ivd} = 0 \qquad i \in \mathcal{N}, d \in \mathcal{D} \tag{5.28}$$

$$h_{op} - \sum_{v \in \mathcal{V}} \sum_{d \in \mathcal{D}} h_{opvd} = 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i \tag{5.29}$$

$$f_{id} - \sum_{v \in \mathcal{V}} f_{ivd} = 0 \qquad i \in \mathcal{N}^B, d \in \mathcal{D} \tag{5.30}$$

In for instance equation (5.23) the arc-flow variable, x_{ijd}, is split by including the vessel index v. Thus, the number of arc-flow variables has now increased by a factor $|V|$.

5.3.3 Decomposed path-flow model

This section presents the reformulated model based on the Dantzig-Wolfe decomposition approach. We have already laid out the basic principles of the decomposition strategy and will now reformulate the original model using a path-flow formulation to break up the problem accordingly.

Interior representation

To formulate a restricted master problem (RMP) we need to make certain modifications to the common constraints, previously represented by (4.5), (4.10)-(4.12) and (4.15).

For each subproblem, corresponding to a vessel and a departure, we need to find a feasible route with regard to the time windows and capacity restrictions while also upholding the restrictions on refueling. A combination of the decision variables in a given subproblem is called a ship route r. For instance, a ship route r for vessel v and departure d includes information about the flow between the nodes i and j, X_{ijvdr}. The new parameters in the path-flow formulation are listed below.

X_{ijvdr}	1 if vessel v travels from i to j on departure d for route r, 0 otherwise
Y_{ovdr}	1 if order o is delivered by vessel v on departure d for route r, 0 otherwise
T_{ivdr}	Departure time from node i for vessel v on departure d for route r
S_{ovdr}	Start time of service for order o by vessel v on departure d for route r
D_{ovdr}	Arrival delay for order o sailing route r on departure d by vessel r
L_{ivdr}	Outbound load at node i for vessel v on departure d for route r
H_{opvdr}	1 if order o is serviced before order p by vessel v on departure d and route r, 0 otherwise
F_{ivdr}	1 if vessel v refuels at node i on departure d and route r,0 otherwise

The constraints are reformulated using convexification before adding them to the RMP. In Section 5.1 we mentioned how DWD uses interior representation to present the common constraints in the RMP. The interior representation of the PSVRSP-RT is shown in constraints (5.31)-(5.38).

$$x_{ijvd} = \sum_{r \in \mathcal{R}_{vd}} X_{ijvdr} \theta_{vdr} \qquad (i,j) \in \mathcal{A}, v \in \mathcal{V}, d \in \mathcal{D} \tag{5.31}$$

$$y_{ovd} = \sum_{r \in \mathcal{R}_{vd}} Y_{ovdr} \theta_{vdr} \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, v \in \mathcal{V}, d \in \mathcal{D} \tag{5.32}$$

$$t_{ivd} = \sum_{r \in \mathcal{R}_{vd}} T_{ivdr} \theta_{vdr} \qquad i \in \mathcal{N}, v \in \mathcal{V}, d \in \mathcal{D} \tag{5.33}$$

$$s_{ovd} = \sum_{r \in \mathcal{R}_{vd}} S_{ovdr} \theta_{vdr} \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, v \in \mathcal{V}, d \in \mathcal{D} \tag{5.34}$$

$$d_{ovd} = \sum_{r \in \mathcal{R}_{vd}} D_{ovdr} \theta_{vdr} \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, v \in \mathcal{V}, d \in \mathcal{D} \tag{5.35}$$

$$l_{ivd} = \sum_{r \in \mathcal{R}_{vd}} L_{ivdr} \theta_{vdr} \qquad i \in \mathcal{N}, v \in \mathcal{V}, d \in \mathcal{D} \tag{5.36}$$

$$h_{opvd} = \sum_{r \in \mathcal{R}_{vd}} H_{opvdr} \theta_{vdr} \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i, v \in \mathcal{V}, d \in \mathcal{D} \tag{5.37}$$

$$f_{ivd} = \sum_{r \in \mathcal{R}_{vd}} F_{ivdr} \theta_{vdr} \qquad i \in \mathcal{N}^B, v \in \mathcal{V}, d \in \mathcal{D} \tag{5.38}$$

A new routing variable θ_{vdr} is introduced, and is equal to 1 if vessel v on departure d chooses to sail route r, and 0 otherwise. Furthermore, $\mathcal{R}_{vd}$ defines the set of all columns in the RMP for vessel v and departure d. The parameter arrays presented are values of the corresponding variables in a corner solution for route r in subproblem (v, d). Not all of these parameters are included in the RMP formulation in the following subsection. However, they are still necessary to describe the underlying conditions of the model. In particular these relations are needed during the implementation phase.

5.3.4 Restricted master problem

The following paragraphs present the objective function and common constraints of the restricted master problem. A complete model formulation is included in Appendix B.

Objective function

$$(RMP)\ \min z = \sum_{v\in\mathcal{V}}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}_{vd}} C^R_{vdr}\,\theta_{vdr} + \sum_{i\in\mathcal{N}^P}\sum_{o\in\mathcal{O}_i} C^X w_o \tag{5.39}$$

The objective function of the RMP is the sum of a convex combination of the total sailing cost C^R_{vdr} from the chosen routes r, and the express delivery cost. The total cost is simply the sum of sailing, fixed and delay costs from the arc-flow model, i.e. C^V_{ijv}, C^F_v and C^P_o respectively.

Common constraints

$$\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}_{vd}} \theta_{vdr} \leq Q_v \qquad v\in\mathcal{V} \tag{5.40}$$

$$\sum_{v\in\mathcal{V}}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}_{vd}} Y_{ovdr}\theta_{vdr} + w_o \geq 1 \qquad i\in\mathcal{N}^P, o\in\mathcal{O}_i \tag{5.41}$$

Constraints (5.40) and (5.41) correspond to (4.11) and (4.15) in the arc-flow model, respectively. As mentioned, these constraints connect the departures by ensuring vessel capacity restrictions and order delivery for all departures.

$$\sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{R}_{vd}} F_{ivdr}\theta_{vdr} - e_i \leq 0 \qquad i\in\mathcal{N}^B, d\in\mathcal{D} \tag{5.42}$$

$$\sum_{i\in\mathcal{N}^B} e_i \leq G \tag{5.43}$$

In constraints (5.42) and (5.43) tanker deployment restrictions are ensured. The constraints correspond to (4.5) and (4.12) from the arc-flow model. Notice that (5.43) do not include a routing variable θ_{vdr}. This is because the deployment variable e_i does not appear in the subproblems.

$$\sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{R}_{vd}} \theta_{vdr} = 1 \qquad d\in\mathcal{D} \tag{5.44}$$

Constraints (5.44) make sure that only one vessel and route is chosen for any departure, and correspond to (4.10) in the arc-flow model. These restrictions may also be named the departure convexity constraints of the RMP.

$$\theta_{vdr} \geq 0 \qquad v \in \mathcal{V}, d \in \mathcal{D}, r \in \mathcal{R}_{vd} \tag{5.45}$$
$$w_o \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{5.46}$$
$$e_i \geq 0 \qquad i \in \mathcal{N}^B \tag{5.47}$$

Finally, non-negativity is imposed on the routing, deployment and express delivery variables. As previously mentioned, the DW method solves the LP relaxation of the MILP, which explains why θ_{vdr} is defined as a continuous variable.

5.3.5 Subproblems

The majority of the constraints from the arc-flow model in Chapter 4 are included in the subproblems. Thus, from the original constraints given by (4.2)-(4.30), only the common constraints are not included in the subproblems. The number of subproblems solved in each iteration is the product of vessels and departures, i.e. $|\mathcal{V}| \times |\mathcal{D}|$.

Each combination of a departure and vessel (v, d) makes up a subproblem. The subproblems are formulated and solved as generalized shortest path problems. For each (v, d) combination we need to find a feasible route for one specified vessel and departure. A feasible route represents a journey originating at the depot at a specified departure time, and travels to a set of platforms to deliver and pickup orders. During a voyage a vessel is required to visit a refueling tanker before returning to the depot. Any buoy is a feasible destination in the subproblem. This is because the final decision regarding tanker deployment is made in the master problem. The route must comply with time windows, sequencing constraints and capacity limitations of the vessel in question. Furthermore, the SP calculates the reduced cost of adding the respective column to the master problem, given by the variable p_{vd}.

Note that the previous vessel choice variable v_{vd} is no longer needed, because each subproblem is solved for a specified vessel. Thus, the final choice of vessel for a given departure is decided in the RMP and incorporated in the routing variable θ_{vdr}. All constraints shown below are illustrated for a single subproblem, i.e. a (v, d) combination. Hence, the SP must be solved for all $v \in \mathcal{V}$ and all $d \in \mathcal{D}$. The variable p_{vd} represents the objective function value of a subproblem corresponding to vessel v and departure d.

Objective function

$$\begin{aligned} (SP)\ \min p_{vd} = &\sum_{(i,j)\in\mathcal{A}} C^V_{ijv}\, x_{ijvd} + C^F_v + \sum_{i\in\mathcal{N}^P}\sum_{o\in\mathcal{O}_i} C^P_o\, d_{ovd} \\ &- \overline{\alpha}_v - \sum_{i\in\mathcal{N}^P}\sum_{o\in\mathcal{O}_i} \overline{\beta}_o\, y_{ovd} - \sum_{i\in\mathcal{N}^B} \overline{\varepsilon}_{id}\, f_{ivd} - \overline{\gamma}_d \end{aligned} \tag{5.48}$$

All cost coefficients from the original arc-flow model are included in the objective function (5.48), except the express cost, C^X. The last four terms represent the respective

RMP dual variables. The dual variables correspond to the common constraints in the RMP. The dual variable $\overline{\alpha}_v$ is linked to constraints (5.40), and $\overline{\beta}_o$, $\overline{\varepsilon}_{id}$, $\overline{\gamma}_d$ relate to (5.41), (5.42) and (5.44), respectively.

Flow network constraints

$$\sum_{j\in\mathcal{N}^P} x_{ijvd} \leq 1 \qquad i \in \mathcal{N} \tag{5.49}$$

$$\sum_{i\in\mathcal{N}} x_{ijvd} = f_{jvd} \qquad j \in \mathcal{N}^B \tag{5.50}$$

$$\sum_{i\in\mathcal{N}^B} f_{ivd} = 1 \tag{5.51}$$

$$\sum_{i\in\mathcal{N}} x_{ijvd} - \sum_{i\in\mathcal{N}} x_{jivd} = 0 \qquad j \in \mathcal{N} \tag{5.52}$$

$$\sum_{j\in\mathcal{N}} x_{0jvd} = 1 \tag{5.53}$$

$$\sum_{i\in\mathcal{N}} x_{i0vd} = 1 \tag{5.54}$$

Constraints (5.49)-(5.54) are equivalent to the corresponding constraints in the arc-flow model except for (4.5), which are included in the RMP. The constraints ensure all flow restrictions between the depot, platform nodes and refueling buoys.

Order balance constraints

$$y_{ovd} - \sum_{i\in\mathcal{N}} x_{ijvd} \leq 0 \qquad j \in \mathcal{N}^P, o \in \mathcal{O}_j \tag{5.55}$$

$$\sum_{o\in\mathcal{O}_j} y_{ovd} - \sum_{i\in\mathcal{N}} x_{ijvd} \geq 0 \qquad j \in \mathcal{N}^P \tag{5.56}$$

Besides constraints (4.15), all order balance constraints are present in the SP. Constraints (5.55)-(5.56) ensure the correlation between the arc-flow variables and the delivery variables.

Order sequence constraints

$$h_{opvd} + h_{povd} - y_{ovd} - y_{pvd} \geq -1 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i \tag{5.57}$$

$$h_{opvd} + h_{povd} \leq 1 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i \tag{5.58}$$

$$h_{opvd} - h_{povd} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i^D, p \in \mathcal{O}_i^P \tag{5.59}$$

The order sequence constraints, (5.57)-(5.59), are all included in the SP. The main purpose of these constraints is to make sure that platforms with several order deliveries during the same journey are assigned an appropriate service sequence.

Time constraints

$$\underline{T}_o \leq s_{ovd} \leq \overline{T}_o + T^M \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{5.60}$$
$$s_{pvd} - (s_{ovd} + T_o^S)\, h_{opvd} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i \tag{5.61}$$
$$t_{0vd} = T_d^D \tag{5.62}$$
$$t_{ivd} - (s_{ovd} + T_o^S)\, y_{ovd} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{5.63}$$
$$s_{ovd} - (t_{ivd} + T_{ijv})\, x_{ijvd}\, y_{ovd} \geq 0 \qquad i \in \mathcal{N}, j \in \mathcal{N}^P, o \in \mathcal{O}_j \tag{5.64}$$
$$t_{jvd} - (T^F + t_{ivd} + T_{ivd}) x_{ijvd} \geq 0 \qquad i \in \mathcal{N}^P \cap \{0\}, j \in \mathcal{N}^B \tag{5.65}$$

Constraints (5.60)-(5.65) are also equivalent to the corresponding constraints in the arc-flow model. Notice that (5.64) and (5.65) are simplified due to the exclusion of the vessel choice variable v_{vd} in the original model. This has certain effect on their linearizations (see Appendix B).

Limit on sailing time

$$t_{ivd} \leq T_d^D - T_{i0}^V + L_v^T \qquad i \in \mathcal{N} \tag{5.66}$$

Time limit for sailing is ensured by constraints (5.66). In this case we no longer need to use linearization due to the exclusion of v_{vd}.

Time delay constraints

$$s_{ovd} - d_{ovd} \leq \overline{T}_o \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{5.67}$$

Constraints (5.67) put a lower bound on the arrival delay variable d_o and are equivalent to constraints (4.26) in the arc-flow model.

Capacity constraints

$$l_{ivd} \leq K_v \qquad i \in \mathcal{N} \tag{5.68}$$
$$\sum_{i \in \mathcal{N}} \sum_{o \in \mathcal{O}_i^D | i \in \mathcal{N}^P} A_o\, y_{ovd} - l_{0vd} = 0 \tag{5.69}$$
$$\begin{aligned} x_{ijvd}\, (l_{ivd} + &\sum_{o \in \mathcal{O}_j^P | j \in \mathcal{N}^P} A_o\, y_{ovd} \\ - &\sum_{o \in \mathcal{O}_j^D | j \in \mathcal{N}^P} A_o\, y_{ovd} - l_{jvd}) = 0 \end{aligned} \qquad (i,j) \in \mathcal{A} \tag{5.70}$$

The capacity constraints, (5.68)-(5.70), are mostly equivalent to the original constraints. However, in (5.68) the vessel choice variable is no longer needed.

Limit on sailing distance

$$\sum_{(i,j)\in\mathcal{A}} B_{ij}\, x_{ijvd} \leq L_v^D \tag{5.71}$$

Constraints (5.71) are the sailing distance constraints for each departure and vessel. The upper limit for maximum sailing distance vary based on the vessel in question. The vessel choice variable is therefore not included.

Variable constraints

The arc-flow and delivery variables in (5.72) and (5.73) are binary. Constraints (5.74)-(5.77) ensure non-negativity for departure time, start of service, delay and load variables. Lastly, binary restrictions are ensured for sequencing and refueling variables in constraints (5.78) and (5.79).

$$x_{ijvd} \in \{0,1\} \qquad (i,j) \in \mathcal{A} \tag{5.72}$$
$$y_{ovd} \in \{0,1\} \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{5.73}$$
$$t_{ivd} \geq 0 \qquad i \in \mathcal{N} \tag{5.74}$$
$$s_{ovd} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{5.75}$$
$$d_{ovd} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{5.76}$$
$$l_{ivd} \geq 0 \qquad i \in \mathcal{N} \tag{5.77}$$
$$h_{opvd} \in \{0,1\} \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i \tag{5.78}$$
$$f_{ivd} \in \{0,1\} \qquad i \in \mathcal{N}^B \tag{5.79}$$

Linearizations and valid inequalities for the decomposed PSVRSP-RT model are included in Appendix B.

Chapter 6

Branch and Price

Branch and Price (B&P) is an important topic in this thesis. The intention of this chapter is to provide an introduction to the terminology and principles of the method, which is used to solve the path-flow model. Section 6.1 introduces the use of column generation for the decomposed model. In Section 6.2 we explain the essential theory of B&P based on related literature. Finally, the B&P scheme used for solving the PSVRSP-RT is presented in Section 6.3.

6.1 Column generation

The path-flow model is solved using column generation based on Dantzig-Wolfe principles. However, the method only finds the optimal solution for the LP relaxation of the PSVRSP-RT. The RMP illustrated in Section 5.3.4 is an LP relaxation due to the continuous non-negativity restrictions imposed on the routing variables θ_{vdr}. For the MILP problem, these variables are binary. The relaxed master problem is in the remainder of this thesis called RMP-LP, while the integer problem is denoted by RMP-IP.

To generate an IP solution to the PSVRSP-RT, a simple heuristic is applied. After the column generation process we are left with a pool of feasible columns. A feasible IP solution may then be found by simply solving the RMP as an IP (RMP-IP) with the existing columns. The heuristic provides a feasible integer solution. In many instances, this method may in fact give a near optimal or even optimal solution, but for large problems this is rarely the case. The problem is that the pool of columns generated for the RMP-LP is incomplete. However, even if the initial RMP-IP solution is non-optimal, it usually provides a relatively strong pessimistic bound.

The reason for applying DW to the PSVRSP-RT is mainly to provide strong bounds for the problem, i.e. RMP-LP giving an optimistic bound and the IP heuristic a pessimistic bound. The column generation process is shown in Algorithm 2.

Algorithm 2: Column generation scheme using (v, d) combinations for the PSVRSP-RT

repeat
  forall the *common constraints in RMP* **do**
    Find the primal solution θ_{vdr} and the corresponding dual solution α_v, β_o and γ_d
  Send the dual variables to the subproblems
  forall the *departures and vessels* **do**
    Solve departure-vessel subproblem
    if *column has negative reduced cost* **then**
      Send new column to RMP
  Solve RMP-LP
until *no new columns with negative reduced cost can be found*;
Solve RMP-IP

In Algorithm 2 the IP heuristic is only applied after the optimal LP solution is found. However, in some instances it might be beneficial to calculate bounds continuously during the iteration process. The IP heuristic is included in the algorithm to illustrate the transition from an the optimal RMP-LP to a feasible solution represented by the RMP-IP, but it is in fact not a part of the column generation scheme.

6.2 B&P methodology

In Branch and Price we combine a Branch and Bound (B&B) search with column generation to guarantee an optimal solution to a large-scale integer program. Before the branching scheme is conducted, the initial root node is calculated using column generation. The subproblems of the decomposed model are termed the pricing problems in the context of B&P.

The set of columns in the root node may be incomplete. Therefore it might not be sufficient to apply a standard B&B approach to the master problem over the existing columns to find the optimal integer solution. Thus, the idea of B&P is to continuously generate new columns while searching the B&P tree. Given a minimization problem, the initial optimal LP relaxation is a lower bound (LB), and placed in the root node. For all other nodes in the B&P tree the pricing problems are solved to find new columns. These columns are used to solve the LP relaxation of the respective node.

6.2.1 B&P algorithm

There are many generalizations of the B&P method depending on best practice for a given model. Desrosiers and Lubbecke (2006) give a good introduction to the basics of B&P and column generation, and we advice the reader to read this paper for further information. Below we have included a few steps that summarize the search procedure:

Step 1 Decide the search strategy, i.e. which node in the B&P tree to solve first. Here there are different strategies, e.g. best-first and depth-first.

Step 2. Impose a branching strategy, i.e. choose which variables to branch on. This also implies imposing all restrictions made by the branching of the specific node, which could affect both the RMP and the SP.

Step 3. Column generation is applied in the respective node of the B&P tree. The RMP is solved with the restrictions imposed in Step 2. Furthermore, the SP is solved using the dual variables from the RMP as well as the new branching constraints. This may give a new column, which is added to the RMP. In this case the RMP is reoptimized. If no column is found, it proves that the current solution of the RMP is optimal.

Step 4. Prune the node if the RMP objective value is not good enough, i.e. violates the current bounds and/or the termination criteria.

Step 5. Check if the RMP solution is integer feasible in the original problem. If so, record the solution and prune the node.

Step 6. Decide on an entity to branch on, i.e. branching strategy. This results in two new nodes, and we continue by redoing the steps.

The procedure above mentions the terms search strategy and branching strategy, which are important characteristics of a B&P algorithm. Depending on the problem structure, an appropriate strategy may have a significant effect on the solution process.

Bounds may also be recorded during the iteration process. This could be helpful to observe convergence and possibly prune nodes. As mentioned, the LP relaxation provides a lower bound (LB) for the problem. In addition, we may also record an upper bound (UB) by solving the RMP-IP with the current pool of columns. This can be conducted in each node during the branching processes. Eventually, the LP solution will converge towards the IP solution, in which case we have found our global optimum.

Many B&P schemes exhibit the so-called tail-off effect (Desrosiers and Lubbecke, 2006). The tail-off effect refers to slow convergence, meaning that there is a small duality gap, but the algorithm struggles to terminate on its own. In this case we may define a termination criteria that accepts the current solution if the duality gap falls below a certain limit.

6.2.2 Branching and search strategies

In the following we use the illustrative model presented in Chapter 5 to explain the basic ideas of a few branching strategies. The example model (P) given by (5.1)-(5.4) represents an outer representation, while the DWD formulation given by (5.15)-(5.22) gives an interior representation.

Looking at the model, an important decision is whether to branch on the original variables x, or the λ variables given in the new formulation. According to Barnhart et al. (1998), branching on the λ variables may create problems along a branch where a variable has been set to zero. The reason is because it puts no restriction on the subproblems, thus the same route may be re-generated with another column indexation. To cope with this problem, we can branch on the original variables. When a variable is set equal to 1, all columns in the master that do not hold this restriction are deleted. In our case this could mean assigning an order to a certain vessel, in which case all columns where the respective order is not assigned to the vessel are excluded from the remainder of the search.

Instead of branching on only one variable, we could use so-called constraint branching, which implies branching on several variables. This strategy has shown good results within practical route planning and normally it results in smaller search trees (Lundgren et al., 2010). Constraint branching is also used in the B&P scheme for the PSVRSP-RT.

After applying a branching strategy and imposing the respective constraints, we get two new nodes in the B&P tree. The next step is choosing the sequence in which we want to evaluate the nodes, i.e. the search strategy. The most common strategies referred to in literature are best-first (BeFS), breadth-first (BFS) and depth-first (DFS), see Clausen (1999). An illustration of typical B&P trees based on these methods are shown in Figure 6.1. The number in each node corresponds to the sequence in which the nodes are solved.

BeFS implies that we evaluate the subproblems that have been created from the node with the most promising optimistic bound. For a minimization problem, this means solving the RMP-LP in each node and choosing the node with the lowest LP bound when moving further down the B&P tree. When using BFS all subproblems are evaluated on a given level in the search tree before the next level is selected and searched. Furthermore, DFS means choosing the subproblem at the lowest level. In Figure 6.1c) DFS always solves the left branch first before we proceed down the tree until an IP solution is found.

In terms of memory requirements, DFS is often more manageable than breadth-first (Clausen, 1999). For BFS the number of nodes at each level of the search tree grows exponentially with the level, making it complicated to do BFS for large problems. In the case of BeFS the memory requirements depends on the solutions of each subproblem, whereas several good solutions may require more memory.

Most of the reviewed literature indicates a more dominant use of BeFS. However, there is no definite intuition on the superiority of either one of the methods. Gunnerud (2011) chooses a best-first strategy as it minimizes the number of nodes solved during the B&P processes. Furthermore, the author points out that we get a more diverse set of solutions early on in the search using BeFS, which is likely to improve the RMP-IP. In the article by Stålhane et al. (2012), a generalized pickup and delivery problem is also solved using a BeFS in the B&P tree. In addition to the three main search methods, it is also common to combine the strategies. For more details regarding search strategies the reader is referred to Clausen (1999).

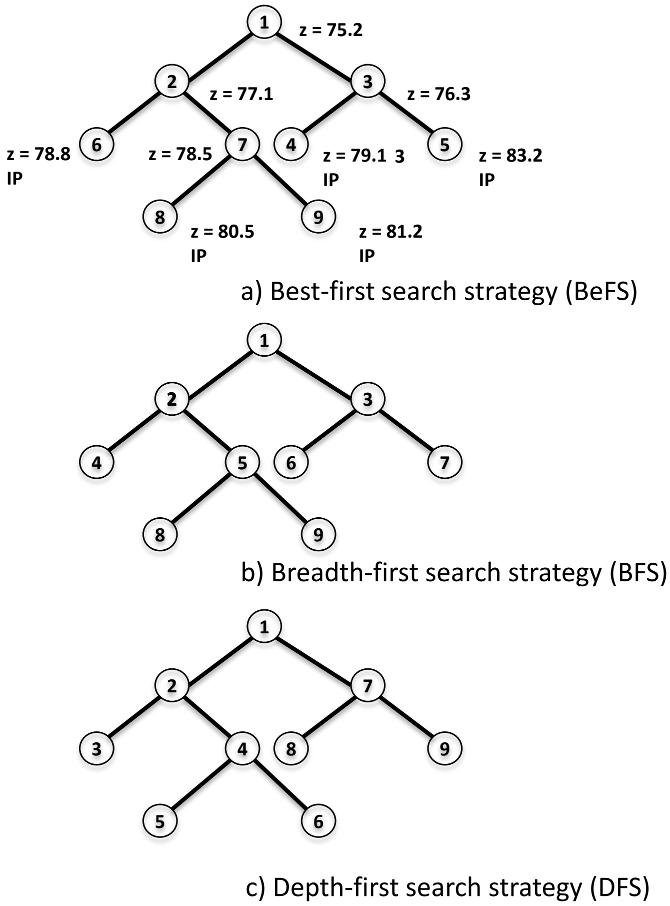

Figure 6.1: *Basic search strategies for B&P trees, based on Clausen (1999)*

6.2.3 Termination criteria

Figure 6.2 illustrates the relation between the lower and upper bounds in a minimization problem (Beasley, 1996). The gap between these bounds is referred to as the duality gap. Essentially, the main objective of a B&P algorithm is to close this gap and prove optimality.

The figure shows the optimistic and pessimistic gaps that point to the difference between the optimal IP solution and the LB and UB, respectively. There is often a trade-off between the bounds in a search algorithm. In some instances it might be more efficient to focus on strengthening the LP bound, while some problems may indicate that the focus should be on improving the best feasible solution. For instance, in a real planning problem where the solver is unable to provide an optimal solution, then closing the pessimistic gap is often the focus. This is because the pessimistic bound represents the current best feasible solution. Thus, a large pessimistic gap means that we have a poor solution.

B&P seeks to evaluate, and possibly rule out, solutions by comparing bounds. To limit the size of the B&P tree it is usually more effective to have strong bounds. On the other hand, the method will also work with weaker bounds. Usually, there is a

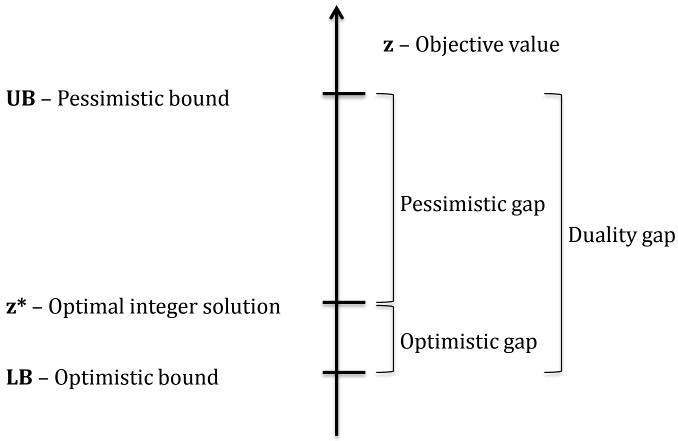

Figure 6.2: *Bounds in a typical minimization problem*

trade-off between the computational efforts associated with computing strong bounds and weaker bounds. Naturally, the first instance will result in smaller trees (Barnhart et al., 1998).

When a node is processed it is not always necessary to solve the RMP-LP to optimality (Barnhart et al., 1998). We could continue branching by accepting the current non-optimal RMP-LP solution in the respective node. The idea of this branching scheme is to provide an integer solution faster by accelerating the search.

A lower bound may be computed by adding the reduced costs to the current RMP-LP value. We know that the sum of the RMP objective function and the reduced costs of the subproblems provides an optimistic bound for the minimization problem, meaning that the node may be pruned if this value is above the global UB (see Algorithm 3). For more information on how to exploit early LP solutions, we advice the reader to view the work of Farley (1990), Vanderbeck and Wolsey (1996) and Barnhart et al. (1998).

Algorithm 3: Termination criteria based on RMP-LP lower bound

repeat
    Solve SP to optimality and obtain $\overline{c}_{SP} < 0$
    if LB_{min} = *RMP-LP* $+\overline{c}_{SP} > UB$ **then**
        Prune the node
    else
        Continue solving RMP-LP
until *no negative columns are found*;

When the main goal is to find a good feasible IP solution and solving to proven optimality is of lesser concern, we can use a greedy branching and searching strategy (Barnhart et al., 1998). This entails branching in a way that makes it more likely to find a good solution in one particular node, and more or less excludes the other. On the other hand, if the aim is to prove optimality, one typically chooses a branching strategy that evenly distributes the solution space between the nodes. The latter case will be emphasized for the PSVRSP-RT in this thesis.

6.2.4 Computational issues

For a minimization problem we know that all columns with negative reduced cost in the pricing problem are candidates to enter the basis. In the context of computational effort, there are different strategies in terms of which columns to add to the RMP. This topic is particularly interesting if the pricing problem is complex.

One approach when the pricing problem is solved is to generate only one column per iteration. For instance, we can add the first column with negative reduced cost, or we can solve the pricing problem to optimality, i.e. add the column with the lowest reduced cost. Clearly, the first alternative will require less computational time per iteration, but it will need more columns to solve the LP of the master problem. The most efficient procedure will depend on the structure of the problem (Vanderbeck and Savelsbergh, 2006).

Barnhart et al. (1998) refers to approximation algorithms as methods to be used when solving the pricing problem. The procedures involve the possibility of generating multiple columns per iteration. A side effect of this approach is an increase in time per iteration, but it could also reduce the total number of iterations needed. However, the overall computational effort from this increase may not be substantial. For instance, it might be easy to find new columns in the pricing problem, thus making it easy to establish a decent pool of columns in the RMP. On the other hand, this approach often leads to an increase in solution time when solving the RMP, due to the rapid increase in columns. For problems where the master problem has a simple structure this might be an acceptable trade-off.

6.3 B&P for the PSVRSP-RT

This section gives a general description of the B&P scheme used to solve the path-flow model introduced in Chapter 5. The challenges of applying B&P is non-trivial, but we have mentioned a few important aspects from the literature that may support the choice of algorithms used for the PSVRSP-RT.

6.3.1 Branching and search strategy

For the PSVRSP-RT we have chosen a strategy that involves branching on the original variables. Another option is to use the routing variables, θ_{vdr}, but this would most likely make the model too restrictive. By original variables we refer to the variables derived from the arc-flow model, e.g. x_{ijvd} and y_{ovd}.

There are a few alternative branching strategies depending on the variable choice. One option is to apply constraints based on the delivery variables, y_{ovd}, which implies fixing orders onto vessels, and then vessels to departures. Another alternative is to use the flow variable, x_{ijvd}, in which case arcs, i.e. (i, j) combinations, are fixed to departures, and then vessels to departures. The most logical choice is to branch on y_{ovd}, which is our chosen strategy. We do this firstly because the delivery parameter, Y_{ovdr}, is present in the master problem, as opposed to X_{ijvdr}. In accordance with Gélinas et al. (1995), this is a good characteristics for branching. Secondly, the intuition is that it is easier to operate with orders as it gives a more practical interpretation of the branching scheme.

Branching procedure

The branching may in fact be seen as two procedures; one where orders are fixed onto departures and the other where orders are fixed to vessels. Below we explain in some detail how the branching scheme is conducted and illustrate with a short example.

Orders onto departures The first step is to check whether an order is divided between several departures. Constraints (6.1) lays out the condition for fractional order deliveries.

$$0 < \sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}_{vd}} Y_{ovdr} \theta_{vdr} < 1 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, d \in \mathcal{D} \tag{6.1}$$

If the condition above holds, then the delivery variable associated with order o is in fact fractional. Constraints (6.1) may return many fractional orders, but we are only interested in the largest fraction. Thus, for each branching iteration we store the order with the largest fraction and name it o^{frac}. Furthermore, we only want two branches, with an even split, which is consistent with the methodology for seeking optimality (Barnhart et al., 1998). Based on the fractional values we need to decide between which departures we should split the solution space. Equation (6.2) can be interpreted as the expected average departure number for the fractional order.

$$\sum_{v \in \mathcal{V}} \sum_{d \in \mathcal{D}} \sum_{r \in \mathcal{R}_{vd}} Y_{ovdr}\, \theta_{vdr}\, d = d^{frac} \qquad o = o^{frac} \tag{6.2}$$

The value d^{frac} refers to the departure number where we wish to insert the split. Since d^{frac} most likely is fractional, we define the branching constraints in the range below $\lfloor d \rfloor^{frac}$ and above $(\lfloor d \rfloor^{frac} + 1)$ for the left and right branch, respectively.

For the left branch, constraints (6.3) and (6.4) are inserted in the master problem (RMP) and subproblems (SP), respectively. Constraints (6.3) ensure that the order o^{frac} is not assigned to neither of the departures $d = 1 \ldots \lfloor d \rfloor^{frac}$. We do so by not choosing any of the columns associated with these delivery variables. Furthermore, constraints (6.4) ensure that the subproblems do not generate columns with the given property.

$$(RMP)\ \sum_{v\in\mathcal{V}}\sum_{d=1}^{\lfloor d\rfloor^{frac}}\sum_{r\in\mathcal{R}_{vd}} Y_{ovdr}\,\theta_{vdr} \leq 0 \qquad o = o^{frac} \tag{6.3}$$

$$(SP)\ y_{ovd} = 0 \qquad o = o^{frac}, v \in \mathcal{V}, d = 1\ldots\lfloor d\rfloor^{frac} \tag{6.4}$$

For all branching constraints we have consciously decided to fix the variables to zero. An alternative would be to claim that the orders must be delivered on a set of departures. However, doing so would exclude the possibility of express deliveries according to constraints (5.41). Thus, the branching would most likely be too restrictive. Additionally, the choice of using zero-constraints is convenient with regard to the implementation process. Since the delivery variable y_{ovd} is fixed to zero for the given range, the resulting dual variables of the new constraints will be zero. Thus, the SP objective function remains unchanged.

In the right branch we use the same procedure as above, only with the remaining range of departures, i.e. $d = \lfloor d_o\rfloor^{frac} + 1\ldots|\mathcal{D}|$. Constraints (6.5) and (6.6) show the right branch insertions to the RMP and SP, respectively.

$$(RMP)\ \sum_{v\in\mathcal{V}}\ \sum_{d=(\lfloor d\rfloor^{frac}+1)}^{|\mathcal{D}|}\ \sum_{r\in\mathcal{R}_{vd}} Y_{ovdr}\,\theta_{vdr} \leq 0 \quad o = o^{frac} \tag{6.5}$$

$$(SP)\ y_{ovd} = 0 \qquad o = o^{frac}, v \in \mathcal{V}, d = \lfloor d\rfloor^{frac} + 1\ldots|\mathcal{D}| \tag{6.6}$$

Example Below we give a short example to illustrate the procedure when branching based on fixing orders onto departures.

Table 6.1: *Determining highest fraction based on LP solution*

Orders/departures	1	2	3	4	5
1	0	1	0	0	0
2	0	0	0	1	0
3	0.25	0	0.25	0.25	0.25
4	0	**0.75**	0.25	0	0

Table 6.1 is a draft of a LP solution that shows how a set of orders may be divided between several departures. According to the procedure, the largest fraction determines which order to use for the next branching scheme, i.e. o^{frac}. The table shows that order 4 holds the largest fraction ($y_{42} = 0.75$).

Table 6.2: *Finding d^{frac}*

Departures	1	2	3	4	5
o^{frac}	0	0.75	0.25	0	0
d^{frac}	1×0	2×0.75	3×0.25	4×0	5×0
	$= 2.25$				

In Table 6.2 we calculate the split ratio, d^{frac}. In this case we get 2.25 which indicates that the split should be between departure 2 ($\lfloor d \rfloor^{frac}$) and departure 3 ($\lfloor d \rfloor^{frac} + 1$).

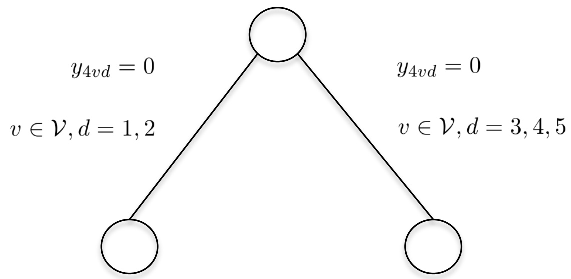

Figure 6.3: *Illustration of the next two branches based on calculated split*

Figure 6.3 illustrates the two new branches in the B&P tree. In addition to the scheme mentioned above, there is a special case involving the express delivery variable, w_o. If an order is split between an ordinary departure and an express delivery, the branching scheme above could result in an indefinite loop. We cope with this aspect by branching on the express variables. The following branches are added to the RMP:

Left branch:

$$(RMP) \quad w_o \leq 0 \qquad\qquad o = o^{frac} \tag{6.7}$$

Right branch:

$$(RMP) \quad w_o \geq 1 \qquad\qquad o = o^{frac} \tag{6.8}$$

The branching constraints (6.7) and (6.8) are only induced as a contingency when there are fractional express delivery variables. Note that they are rather simple as they only influence the master problem, therefore no constraints are added in the SPs.

Orders onto vessels The second branching procedure is to ensure that all orders are assigned to a single vessel. Constraints (6.9) are equivalent to (6.1), except now we look at fractional vessel assignments.

$$0 < \sum_{d \in \mathcal{D}} \sum_{r \in \mathcal{R}_{vd}} Y_{ovdr}\, \theta_{vdr} < 1 \qquad\qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, v \in \mathcal{V} \tag{6.9}$$

As before, order o associated with the largest fraction is set as o^{frac}. Furthermore, the branching split is decided for the vessels by calculating v^{frac}. This procedure is shown in equation (6.10).

$$\sum_{v \in \mathcal{V}} \sum_{d \in \mathcal{D}} \sum_{r \in \mathcal{R}_{vd}} Y_{ovdr}\, \theta_{vdr}\, v = v^{frac} \qquad o = o^{frac} \tag{6.10}$$

The branching scheme is conducted as the previous procedure. Constraints (6.11)-(6.12), and (6.13)-(6.14) show the new restrictions for the left and right branch, respectively.

Left branch:

$$(RMP) \sum_{v=1}^{\lfloor v \rfloor^{frac}} \sum_{d \in \mathcal{D}} \sum_{r \in \mathcal{R}_{vd}} Y_{ovdr}\, \theta_{vdr} \leq 0 \qquad o = o^{frac} \tag{6.11}$$

$$(SP)\ y_{ovd} = 0 \qquad o = o^{frac}, v = 1 \ldots \lfloor v \rfloor^{frac}, d \in \mathcal{D} \tag{6.12}$$

Right branch:

$$(RMP) \sum_{v=(\lfloor v \rfloor^{frac}+1)}^{|\mathcal{V}|} \sum_{d \in \mathcal{D}} \sum_{r \in \mathcal{R}_{vd}} Y_{ovdr}\, \theta_{vdr} \leq 0 \qquad o = o^{frac} \tag{6.13}$$

$$(SP)\ y_{ovd} = 0 \qquad o = o^{frac}, v = \lfloor v \rfloor^{frac} + 1 \ldots |\mathcal{V}|, d \in \mathcal{D} \tag{6.14}$$

Algorithm 4 summarizes the branching strategy for the PSVRSP-RT. The procedure ends when branching is done and we have created two new nodes. The Branch and Price scheme continues by solving the next node based on the chosen search strategy.

Algorithm 4: Branching strategy for the PSVRSP-RT

```
forall the orders and departures do
    if a routing variable splits an order between several departures then
        Find highest fraction, o^frac
        if an express variable is greater than zero, i.e. w_{o^frac} > 0 then
            Create left branch: w_{o^frac} ≤ 0
            Create right branch: w_{o^frac} ≥ 1
        else
            Calculate branching split, d^frac
            Create left branch - (1 ... ⌊d⌋^frac)
            Create right branch - (⌊d⌋^frac + 1 ... |D|)
            Add new nodes to list of unsolved problems
if no order is split on several departures then
    forall the orders and vessels do
        if a routing variable directs orders on several vessels then
            Find highest fraction, o^frac
            Calculate branching split, v^frac
            Create left branch - (1 ... ⌊v⌋^frac)
            Create right branch - (⌊v⌋^frac + ... |V|)
            Add new nodes to list of unsolved problems
if an order is neither split on several departures or several vessels then
    Solution is integral
```

As the algorithm points out, a node will only consider branching on orders onto vessels once there are no more departure splits in the RMP solution. However, after ensuring the integrality property for orders onto departures, the solution is likely to be integer for orders onto vessels as well. Thus, the last procedure may be regarded as more of a control sequence since the vessel choice most likely is resolved implicitly during previous branching stages.

This branching strategy will in most instances guarantee that integer requirements for the original variables hold. However, a special case may arise where the model chooses more than one geographical route associated with the same orders, vessel type and departure. For this to happen the routes must have the same total cost. In this case either route will provide a satisfactory solution and we prove optimality by simply choosing one of them.

Search strategy

For the PSVRSP-RT we have chosen a best-first search strategy (see Figure 6.1). Early computational results showed that the model provides a strong LP bound and the corresponding branching scheme lead to a relatively small B&P tree. The results also showed that the optimal solution often is found in the root node, which forced the program to search all nodes regardless of strategy. For this reason the choice of search strategy will most likely not make much difference. BeFS was chosen mostly based on reviewed literature, but it can also be argued that it is more practical in terms of implementation.

6.3.2 Algorithm

An overall representation of the B&P scheme is shown in Algorithm 5. The pseudo-code combines the branching and search strategy with column generation to find the optimal solution.

Algorithm 5: Branch and Price for the PSVRSP-RT

```
forall the (v, d) combinations do
    Generate initial columns
Create root-node
repeat
    Choose next node based on best-first search strategy
    repeat
        Solve RMP
        Send dual values for common constraints to subproblem (SP)
        forall the (v, d) combinations do
            Solve SP
            if column has negative reduced cost then
                Send new column to RMP
    until no new columns with negative reduced cost is found;;
    Solve RMP-IP
    Update bounds
    if Duality gap < termination criteria then
        Optimal solution is found
    else
        if node is feasible and LP < global UB then
            Branch and create new nodes
            Add new nodes to list of unsolved problems
until optimal solution is found;
```

Although the algorithm above gives an idea of the B&P procedure used for the PSVRSP-RT, it does not describe all aspects. We mentioned earlier that there are various ways to generate columns. It is not always necessary to solve an SP to optimality and it might be beneficial to add multiple columns to the RMP in each iteration. Knowing what is the preferred choice is not trivial and must be tested for any given problem instance.

Furthermore, different termination criteria may also be introduced (Barnhart et al., 1998). The algorithm states that the model terminates whenever the IP gap is significantly small, but there are also other alternatives. For instance, the RMP-LP does not necessarily have to be solved to optimality. As mentioned in Section 6.2.3, if the minimum lower bound of a node is above the current upper bound, then the node may be pruned. This requires the subproblems to be solved to optimality to obtain the LB_{min} in accordance with Algorithm 3.

In Algorithm 5 we observe that the IP heuristic is calculated for every iteration. Naturally, this is not necessary for all iterations, but by continuously solving the RMP-IP we get a steady improvement of the upper bound, which is essential with regard to premature termination.

Overall, it is important for the reader to understand that the B&P scheme used in this thesis was formed based on a substantial amount of testing. Thus, the final algorithm is non-trivial and therefore more suited to be illustrated in practice rather than theory. These topics are discussed in more detail in the computational study (see Chapter 8).

Chapter 7

Implementation

This chapter covers the implementation of the arc-flow and path-flow model formulations with commercial optimization software. The models have been implemented in the Mosel programming language using the Xpress-MP system. Section 7.1 briefly presents Xpress-MP and the hardware utilized for solving the models. Section 7.2 goes through some of the pre-processing conducted in Xpress-MP which is common for both models. In Section 7.3 modeling choices for the arc-flow model is covered. Section 7.4 presents the implementation of the DWD path-flow model and specifics for the DW algorithm. Section 7.5 goes further into the implementation of Branch and Price for the path-flow model. Solving the optimization models requires proper input data, which in this thesis approximates the current situation in Macaé. Section 7.6 presents the input data that acts as a basis for later computational analyses.

7.1 Software and hardware

Mosel provides a language that is both a modeling and a programming language (Xpress-MP User's Manual, 2006). Input to Mosel is a .mos file that contains the model to be solved. This file is compiled, which results in a binary model (BIM) that is saved as a .bim file. The model is run when Mosel reads the BIM file and executes it. Mosel does not integrate a solver by default, but offers connections to external solvers through modules. The module *mmxprs* connects Mosel to the Xpress-Optimizer, through the procedure *minimize* that optimizes the problem at hand. The Xpress-Optimizer is designed for solving linear, mixed integer and quadratic programming problems. The module *mmjobs* extends the Mosel language by allowing communication between .bim files. It is used here to allow parallel solving of subproblems in the path-flow model.

The Mosel programming environment may be accessed through Xpress-IVE, the Xpress-MP integrated visual environment. The Xpress-MP system utilizes the simplex and Branch and Bound algorithms to solve problems. In addition, a powerful presolver tightens the problem to solve it more efficiently by removing redundant constraints and variables. The Xpress-MP presolver may even recognize and create special ordered sets and semi-continuous variables, without being specified explicitly by the user (Ashford, 2007). The path-flow model in this thesis uses Branch and Price to optimize the problem. The B&P algorithm has been coded directly in the Xpress-MP environment through Xpress-IVE.

The models have been implemented and solved using FICO Xpress Mosel 64-bit version 3.4.0, which we term Xpress-MP in the remainder of this thesis. The programs are run on a cluster containing 41 independent nodes with the following specifics: HP dl165 G6, 2 x AMD Opteron 2431 2.4 GHz, 24 Gb RAM and 164 Gb SAS 15000rpm.

7.2 Pre-processing in Xpress-MP

In addition to the built in presolver, some manual pre-processing of the input data is common for both the arc-flow and path-flow models. In Xpress-MP, dynamic arrays are used for variables and constraints. Xpress-MP handles dynamic arrays efficiently and does not use extra memory or processing power when an index in a dynamic array is empty (Xpress-MP User's Manual, 2006). For instance, an arc variable in the arc-flow formulation x_{ijd} is initialized from the set $(i, j) \in \mathcal{A}$, in which $\mathcal{A}$ contains all arcs between the depot, offshore installations and buoys. Since only one buoy can be visited on a voyage, no arcs between buoys are created. That is,

$$x_{ijd} = 0 \qquad (i, j) \in \mathcal{N}^B \times \mathcal{N}^B \tag{7.1}$$

The same is true for arc variables for a single node,

$$x_{iid} = 0 \qquad i \in \mathcal{N} \tag{7.2}$$

and sequencing variables for a single order,

$$h_{ood} = 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{7.3}$$

A scenario arises in the computational study in which all vessels must refuel at the end of a voyage. This is done in Xpress-MP by creating all arcs going into the buoys as they otherwise would, and skipping the variable declarations for arcs between bouys and the offshore installations. That is,

$$x_{ijd} = 0 \qquad i \in \mathcal{N}^B, j \in \mathcal{N}^P \tag{7.4}$$

Data input files are identical for the arc-flow and path-flow models. The distance matrix B_{ij} is not explicitly given in this data file. From a set of coordinates for every node, B_{ij} is calculated at the start of the program from the formula

$$B_{ij} = \sqrt{(a_i - a_j)^2 + (b_i - b_j)^2} \qquad (i,j) \in \mathcal{N} \tag{7.5}$$

Here, a_i and b_i represent the x- and y-coordinates in a two-dimensional coordinate system for node i, respectively. For a detailed description of all pre-processing of variables, constraints and input data, we refer to Appendix C which includes the .mos files for running the Xpress-MP programs.

7.3 Arc-flow model

We refer to the arc-flow model formulation as the objective function (4.1), all constraints (4.2)-(4.42) and linearizations (4.43)-(4.55). This model is implemented and solved in Xpress-MP as a mixed integer linear program (MILP). Branch and Bound is used to find the optimal solution of the MILP. Solving MILPs is generally harder than solving equivalent linear programs. The solver utilizes the structure of the model and its built in presolver to valid inequalities before the B&B algorithm is initiated. The B&B algorithm systematically goes through the solution space while discarding subsets of solutions that are infeasible or non-optimal.

Valid inequalities that narrow the solution space of the LP relaxation, yet still cover the entire solution space of the MILP, can enhance the B&B algorithm in discarding nodes more quickly (Kallehauge et al., 2005). Adding valid inequalities on the other hand increases the number of constraints, which may be a complicating factor. The influence of valid inequalities on the solution process was analysed in Friedberg and Uglane (2012b), which concluded that constraints (4.56)-(4.62) gave the most promising results to the test cases that were run. These are therefore also implemented in the arc-flow model. Worth noting is that the subtour elimination constraints in (4.57)-(4.58) eliminate subtours in sets containing up to 4 nodes in the arc-flow model.

Generating lower and upper bounds

The B&B algorithm generates an optimistic and a pessimistic bound on the optimal objective value while it searches for the optimal solution. The pessimistic upper bound on the optimal objective value is the best solution that has been found up to a given point in the tree search. At the root node of the B&B tree, after the presolver is done, the LP relaxation generates a lower bound on the optimal objective value. This duality gap that arises between the lower and upper bounds was illustrated in Figure 6.2 in Chapter 6. The lower bound is raised as the solver searches through the tree and better LP relaxations can be found, while the upper bound is lowered when branching leads to better IP solutions.

7.4 Path-flow model

We refer to the path-flow model as the restricted master problem, given by constraints (5.39)-(5.47), and the subproblems, given by constraints (5.48)-(5.79). Linearizations of the subproblems can be found in Appendix B. This model is implemented in Xpress-MP through two separate program files, one for the RMP and one for the SPs. The RMP program file contains Dantzig-Wolfe algorithms for finding the optimal linear program solution. Branch and Price algorithms for closing the duality gap that may arise after the DW algorithm is finished are also included in the RMP file, and will be further discussed in Section 7.5. This section thus focuses on the process of finding the root node in the B&P tree.

7.4.1 Restricted master problem

When decomposing based on departures the RMP chooses a PSV and a voyage for each departure within the time horizon considered. In addition, express delivery of orders and geographical placement of refueling tankers are chosen. The Dantzig-Wolfe algorithm finds optimal solution to a linear program, which means that the restricted master problem (RMP) must be solved as an LP to obtain dual values that can be communicated to the subproblems, as was discussed in Chapter 5.

The routing variables, θ_{vdr}, are initialized as integer. This is to enable the IP heuristic at the end of the DW algorithm. The RMP is solved using the procedure *minimize(XPRS_ LIN,RMP)*, which changes all integer variables into linear, in order to solve the LP relaxation of the problem. The Xpress-MP presolver does not function properly with dual information, therefore the presolver must be shut down before solving the RMP. All constraints and variables are initialized and finalized before the first iteration of the RMP, to save computational time between iterations.

Initial columns

Initial columns must be created such that the RMP finds a feasible solution and thereby feasible dual values in the first iteration. The heuristic for finding initial columns ensures the following:

- One route is generated for each vessel and departure
- No orders are serviced on these routes
- Only the closest refueling buoy is visited

An initial voyage only goes from the depot to a refueling tanker, and back to the depot. Since no orders are serviced on these routes, all orders must be sent by express vessels. The PSVs visit the same refueling buoy to ensure feasibility with respect to the number of refueling tankers deployed. These routes are clearly not close to the optimal routes due to the express deliveries. The column generation that follows is therefore expected to provide significant reduced costs in all subproblems.

Other ways of obtaining initial columns for faster convergence may be evaluated, but are not considered here.

Generating an upper bound with the IP heuristic

After all necessary columns have been added to the RMP, and no other columns with negative reduce costs can be obtained, the optimal LP solution of the master problem has been found. This gives a lower bound to the master problem. To obtain a feasible integer solution, the RMP is solved once more as a MILP with the same columns, using simply the procedure *minimize(RMP)* with the routing variables now being integer. This heuristic gives an upper bound for the master problem, and hopefully an objective value close to optimality. Furthermore, the column generation terminates and the root node of the B&P tree has been solved. We may end up with a MILP gap, given that the upper and lower bounds are different.

To generate upper bounds on the optimal objective value, the IP heuristic may be utilized before all necessary columns have been added to the RMP since

$$z^* \leq z^{IP} \tag{7.6}$$

Here, z^* represents the optimal objective value, and z^{IP} is any feasible IP solution. Solving the RMP is computationally rather simple compared with solving the subproblems, which means that this is a cheap way of obtaining more information on the optimal objective value before the root node has been solved. In addition to generating an upper bound, the z^{IP} constitutes the best solution to be found at that time, which may prove beneficial for an operational optimization model that requires good solutions quickly.

7.4.2 Subproblems

The subproblems are simply smaller versions of the original arc-flow model. One SP is equivalent to an arc-flow model consisting of one departure and one vessel type. The solution processes are therefore identical in Xpress-MP. The solver utilizes B&B to systematically find integer solutions to the SPs. Since the presolver was shut down in the RMP, the presolver must be turned on in each SP before it is solved.

Valid inequalities are included to enhance the B&B tree search in the SPs. Since the SPs in the DWD model are smaller versions of the original arc-flow model, we have included smaller sets of subtour elimination constraints, (4.57)-(4.58). For the SPs the constraints eliminate subtours of sets containing only 2 nodes, due to the trade-off between adding cuts and increasing the number of constraints. Valid inequalities in constraints (4.59)-(4.62) are also included (see Appendix B for a full description of valid inequalities in the SPs).

Furthermore, all constraints and variables are initialized and finalized the first time the SPs are loaded, in order to save computational time when they are executed thereafter.

Adding columns to the RMP

There may be many solutions with negative reduced costs given specific dual information from the RMP, as was discussed in Chapter 6. The most promising one is the optimal solution to the subproblem. However, any column with negative reduced cost is expected to improve the solution of the RMP when included. The problem considered in Friedberg and Uglane (2012b) experienced long solution times to close relatively small MILP gaps. The problem of this thesis, apart from the additional placement of refueling tankers, is equivalent. Therefore, to enhance the solution process, the B&B tree search can be stopped as soon as the subproblem considered has returned a column with negative reduced cost. At least during the first few iterations of the DW algorithm this may prove helpful, since all columns added is expected to drastically improve the RMP objective value. This might guide the RMP more quickly towards the optimal solution. Of course, this approach may exclude better routes that are within close proximity of the one accepted, especially as the RMP moves closer to the optimal solution. A combination of the two is also a possibility, for instance by accepting quick solutions early in the process, while focusing more on finding good solutions to subproblems after a number of iterations.

Adding more columns at once is another possibility. This is to exploit the fact that many routes in the subproblems may contribute to a better description of the solution space in the RMP. The RMP is easier to solve than the subproblems, which means that adding multiple columns may contribute to a faster convergence if the total number of iterations required decreases. These approaches are further analysed in Chapter 8.

Generating a lower bound

A benefit of solving the subproblems to optimality is that it also allows us to create a lower bound on the optimal objective value. This dual bound is obtained by adding the reduced cost from the most promising vessel for each departure to the objective value of the RMP.

$$z^* \geq z^{RMP-LP} + \min_{v \in \mathcal{V}}\{\bar{c}_{v1}\} + \min_{v \in \mathcal{V}}\{\bar{c}_{v2}\} + \ldots + \min_{v \in \mathcal{V}}\{\bar{c}_{v|\mathcal{D}|}\} \tag{7.7}$$

In (7.7) $\bar{c}_{vd}$ is the optimal objective value of subproblem (v, d), i.e. its reduced cost. z^* is the optimal LP objective value, and z^{RMP-LP} is the RMP-LP objective value at the current iteration. The reasoning behind this lower bound is that a column may improve the objective value of the RMP by at most its reduced cost. Only one vessel may be used at any departure, therefore using the most promising one at each departure gives an optimistic bound. For more information on this topic, we advise the reader to view Karlof (2006). As with an upper bound, a lower bound reveals more information about the optimal objective value before the DW algorithm has finished, which may prove helpful.

7.4.3 Solving subproblems by parallel computing

Gunnerud (2011) states that the greatest advances in computing power have the last few years come through an increase in the number of CPUs (Central Processing Unit) in a computer, and not as a result of an increase in CPU clock speed. Parallelization is a way of utilizing computing power more efficiently through solving different parts of the program on separate CPUs simultaneously. Tebboth (2001) utilizes parallel computing to significantly reduce solution times on a DWD problem. The work was based on the parallel DW algorithms that are introduced in Ho et al. (1988). Gunnerud et al. (2010) obtain a speedup of around 2.7 on a DWD petroleum production problem by the use of parallel computing.

The path-flow model is decomposed into $|\mathcal{V}| \times |\mathcal{D}|$ independent subproblems that need to be solved in each iteration. The block angular structure of the DWD problem suggests there is a potential for reducing solution times by means of parallel computing.

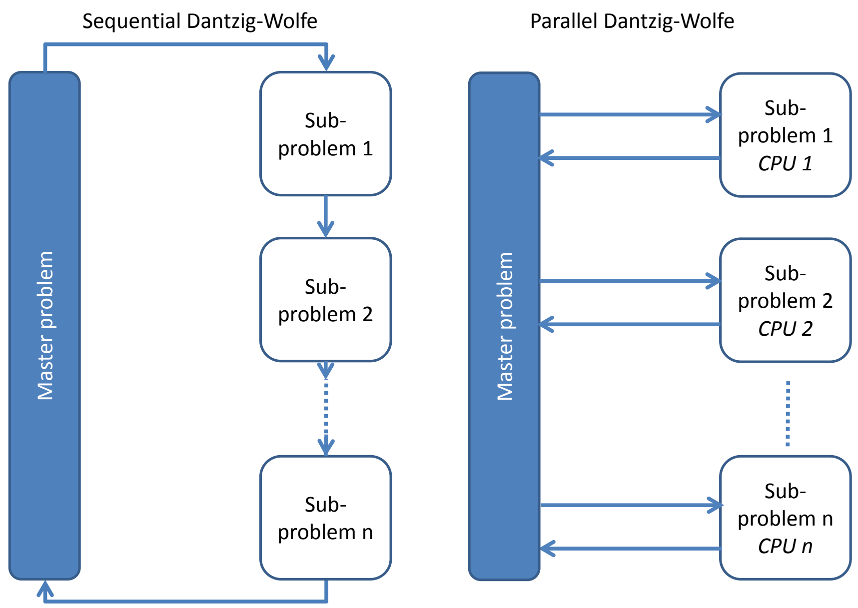

Figure 7.1: *Sequential versus parallel Dantzig-Wolfe solution algorithm. In the parallel DW, the RMP is solved when all subproblems have given a signal that they are finished (barrier synchronization).*

The idea of parallel computing is to assign one core of the CPU to each subproblem, such that the independent subproblems are solved simultaneously rather than in sequence. An illustration of this concept is given in 7.1. A multi-core CPU, which is a processor containing multiple CPUs, has the potential of reducing the computing time during one iteration by a factor equal to the number of cores, not taking into account the time required to solve the restricted master problem. However, the reduction is dependent on various factors. For instance, assuming the number of subproblems solved on a multi-core CPU computer is $|\mathcal{V}| \times |\mathcal{D}|$, the RMP cannot be solved until the most computationally costly subproblem is solved. Thus, as some subproblems finish before others, their respective processors remain idle until the next iteration. This basic form of parallel computing is termed barrier synchronization (Foster, 1995). The CPUs run until they have reached their barrier, for instance when optimal solutions to their respective subproblems have been found. A more sophisticated method for parallelization of a DWD problem, termed accelerated feedback, is considered in Gunnerud (2011). In this thesis we consider the basic barrier synchronization only.

Communication between the RMP and SP

To enable parallel solving of subproblems, the module *mmjobs* is included in Mosel. This module lets Mosel establish a communication between several models in separate program files (Xpress-MP User's Manual, 2006). All subproblems are compiled, followed by the creation of a .bim file. After being loaded, the SPs await a signal, an event, from the RMP file before they are executed. The shared memory IO driver is used to send information from one file to many (Colombani and Heipcke, 2006). RMP dual values are sent to all SPs via the shared memory IO driver. After the RMP has been solved to optimality, it sends execute events to all SPs and wait for signals from all SPs that they are finished before re-solving with the new columns. An SP can either return new columns, return a signal that no new columns were found, or return a signal that something went wrong during the execution.

The new columns are sent from the SPs through the memory pipe IO driver, which is used to send information from many files to one. A column describes an entire route for a vessel on a given departure, and all parameters are sent through the memory pipe IO driver. In addition, if the SP is solved to optimality, the objective value, i.e the reduced cost, is sent to the RMP in order to calculate a lower bound on the optimal MP objective value. For more information on communication between multiple files in Xpress-MP we refer to Colombani and Heipcke (2006).

7.5 Branch and Price

This section covers the implementation of Branch and Price for the path-flow model in Xpress-MP. Implementing B&P requires coding of the algorithms, as they are not built-in functions of the Xpress-MP system such as simplex and Branch and Bound. For a complete list of choices made, we refer to the program files in Appendix C.

The B&P search tree

After the root node has been solved and a fractional solution remains in the RMP-LP, two nodes are created based on the branching strategy that was discussed in Chapter 6. The root node is marked finished when two new nodes are created, and is not considered in the remainder of the search tree. Both nodes are solved using column generation with the proper constraints added in both the RMP and subproblems.

The left branch is always solved first. Each node is given a *record* in Xpress-MP that stores all necessary information for that node, including a pointer to its parent node, overview of constraints put on its branch and its optimal LP solution after being solved.

An array of all B&P constraints is stored. Another array states which of these constraints are induced on the RMP in each node of the tree. These arrays are used in combination with the procedure *sethidden* in Xpress-MP to hide or unhide the correct constraints when a node is being solved. Furthermore, all columns are included in the pool of available columns when the RMP is solved, yet many of these columns are infeasible due to the constraints that are unhidden using the *sethidden* procedure.

Bounds and termination criteria

As the search continues in the B&P search tree, a large number of routes is created in each node. These are feasible routes in the original path-flow formulation, which means that they help describe the solution space of the master problem. To utilize the great increase in feasible routes, the IP heuristic may be initiated as the search progresses. In the code implemented in Xpress-MP, the user may therefore specify a number of nodes to be searched before the IP heuristic is initialized and hopefully creates a new global upper bound. Since all columns added during the B&P process are feasible in the original problem, the *sethidden* procedure hides all constraints that were put on the node that was solved before initiating the IP heuristic. The RMP is computationally easily solved compared with the SPs, therefore this is a cheap way of utilizing the great amount of columns to enhance the search.

The pessimistic bound is the currently best IP solution. If the IP heuristic finds a better upper bound, all nodes that are marked as unfinished are checked and pruned if their LP value given a specific tolerance is above the new upper bound. New upper bounds may also be found by branching until an IP solution is found, which initiates the same check of all unfinished nodes. Naturally, these nodes could never provide a better IP solution than the global upper bound, knowing that

$$z^{LP} \leq z^{IP} \tag{7.8}$$

for any node.

When the right and left branches of the current node has been solved, the optimistic bound on the optimal objective value is the node with the lowest LP value, which has not yet been branched on. The best first strategy (BeFS) says that this node is the next to be branched on as the B&P algorithm continues. The algorithm terminates when this lower bound is within the specified MILP gap tolerance. It also terminates if there are only integral solutions left.

Worth noting is that a node can never lead to an infeasible solution. This is ensured because the chosen branching scheme never excludes express delivery. If the node leads the model to choose an express vessel, the objective value is expected to become rather high in the LP relaxation, which suggests pruning. However, without using equation (7.7), the node has to be entirely solved before it can be pruned. Using the equation when subroblems are solved to optimality therefore comes with another termination criterion. Pruning is allowed if the right hand side of equation (7.7) for that node becomes larger than the global upper bound, given the specified tolerance. This termination criterion was described in Algorithm 3 in Chapter 6.

Adding constraints to subproblems

As described in Chapter 6, the subproblems are imposed with the same constraints as the RMP with respect to delivery of orders. To ensure that the SPs include the correct constraints, the node *record* contains arrays of the orders that have been branched on, whether they are imposed on departures or vessels. These arrays, called *ZeroMatrices* in the code, are sent through the memory pipe IO driver before the SPs are solved. Using the procedure *sethidden*, the correct y_{ovd} variables are forced to zero in the SPs. This logic is illustrated below by a continuation of the example from Chapter 6, shown in Table 6.1.

Example The example branches on order 4 and departure 2. Let us assume this branching was conducted in the root node (node 1). The *ZeroMatrix* for orders onto departures which is then sent to the subproblems in node 2, the left branch, can be seen in Table 7.1. The value 0 indicates that order 4 cannot be serviced on departures 1 and 2. A similar matrix is sent for orders onto vessels, which in this case are all non-zero, since no branch has been imposed on vessels. The *ZeroMatrix* for orders onto departures in the right branch, node 3, is shown in Table 7.2.

Table 7.1: *ZeroMatrix for orders onto departures sent to the subproblems in node 2 (left branch) of the B&P search tree*

Orders/departures	1	2	3	4	5
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	1	1
4	0	0	1	1	1

Table 7.2: *ZeroMatrix for orders onto departures sent to the subproblems in node 3 (right branch) of the B&P search tree*

Orders/departures	1	2	3	4	5
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	1	1
4	1	1	0	0	0

Before the SPs are run, constraints (7.9) are imposed on them. They state that no orders can be serviced.

$$y_{ovd} = 0 \qquad\qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, v \in \mathcal{V}, d \in \mathcal{D} \tag{7.9}$$

The *ZeroMatrix* then indicate with the value 1 which of these constraints that are to be hidden before the subproblems are solved. Constraints (7.9) are therefore reduced to

$$y_{4vd} = 0 \qquad\qquad v \in \mathcal{V}, d = 1, 2 \tag{7.10}$$

in the left branch (node 2), while they in the right branch (node 3) are reduced to

$$y_{4vd} = 0 \qquad\qquad v \in \mathcal{V}, d = 3, 4, 5 \tag{7.11}$$

7.6 Input datasets

The mathematical models presented in Chapters 4 and 5 need realistic data input to be value adding. In this section the most important input parameters used in later computational analyses are presented. The aim of the input parameters is to simulate the current state of Logistics in Macaé, which primarily serve facilities in Campos basin. In the end of the section, seven cases are constructed based on model assumptions and the real world parameters. The parameters are based on Friedberg and Uglane (2012a), Friedberg and Uglane (2012b) and discussions with planners in Macaé during Spring 2013. Certain assumptions behind the mathematical model formulation imply that simplifications have been made.

Geographical placement of facilities and buoys

Figure 7.2 shows the placement of 20 offshore facilities in Campos basin. Most of the facilities are production platforms, which imply that these positions are fixed for a long period of time. The facilities are chosen from different production fields that are currently operational in Campos; Marlim, Marlim Sul and Congro, among others.

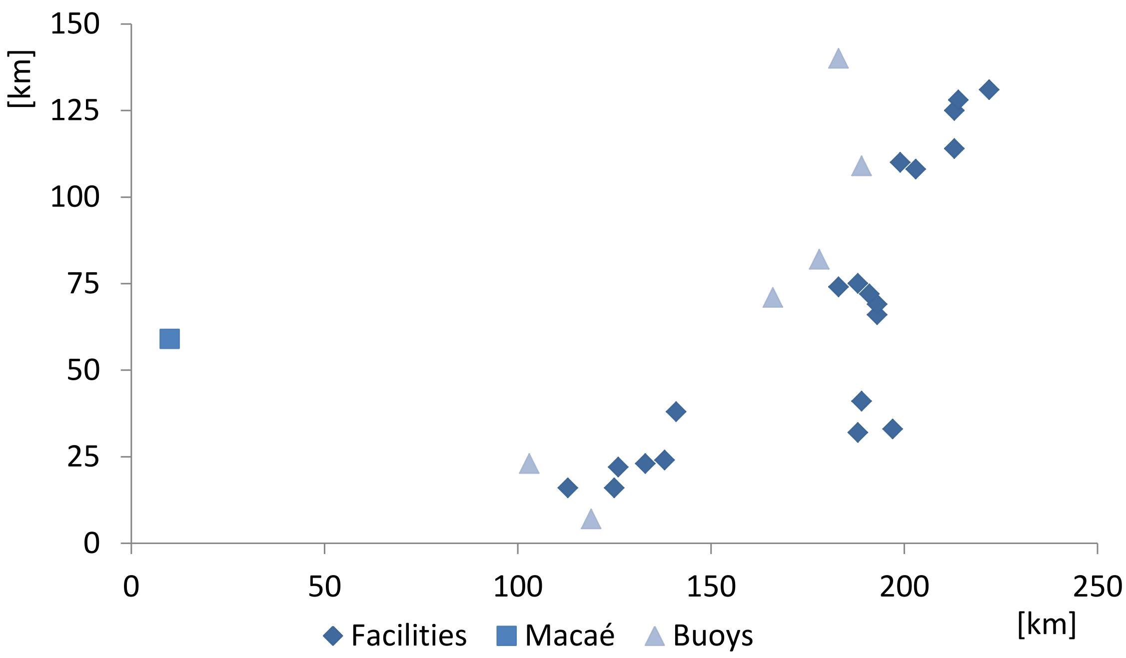

Figure 7.2: *Geographical placement of facilities and buoys in Campos basin*

For a complete list of facilities and their corresponding placement in the figure, see Table 7.3. In addition to the offshore facilities, there are six buoys for deploying refueling tankers. Macaè is located approximately 110 kilometers from P-12, its nearest offshore facility. The distance from Macaé to the facility farthest away, P-52, is approximately 220 kilometers.

Table 7.3: *Placement of facilities and buoys in Campos basin, coordinates given in [km] on x- and y-axis, respectively*

Facility	x	y	Buoy	x	y
Macae	10	59	BE-1	103	23
P-12	113	16	BE-2	178	82
P-07	125	16	BE-3	189	109
P-65	126	22	BE-4	183	140
P-15	133	23	BE-5	166	71
P-08	138	24	BE-6	119	7
P-09	141	38			
P-51	188	32			
P-56	197	33			
P-40	189	41			
P-26	193	66			
P-18	193	69			
P-19	191	72			
P-20	188	75			
P-27	183	74			
P-25	203	108			
P-52	222	131			
P-54	213	125			
P-50	213	114			
P-31	199	110			
FPSO-BR	214	128			

Costs

The objective function (4.1) in the arc-flow model formulation minimizes total costs over the planning period. These costs can be divided into two groups, direct and indirect costs. The direct costs are further split into variable and fixed sailing costs, which can be seen in Table 7.4.

Table 7.4: *Direct costs for the PSV 1500 and the PSV 3000*

	Variable [per km]	Fixed [per voyage]
PSV 1500	\$ 280	\$ 87,500
PSV 3000	\$ 420	\$ 122,500

The fixed costs are based on daily lease rates for PSV 1500 and PSV 3000 of \$ 25,000 and \$ 35,000, respectively. A vessel is normally utilized for two voyages per week, thus the lease rates are multiplied by a factor of 3.5. The variable costs mainly consist of fuel that is supplied by Petrobras, in addition to other operating costs that are in most contracts paid by the third party operator. Variable costs are set such that there is a balanced relation to the fixed costs.

The indirect costs consist of delay penalty for late delivery of an order, and the express penalty for assigning an order to an express vessel. The express delivery cost is set at a high symbolic value, thus forcing the model to use express as a last resort. It is difficult to measure the cost of delay. Delivering an order too late might interfere with other planned operations, e.g. maintenance. The worst-case scenario may be production shutdown, which is a great concern for the platform managers offshore. Logistics in Macaé face problems with orders being delivered overdue, which lead to a lack of trust in the entire logistics operations. Delay is therefore penalized by a large factor to enforce on-time delivery. The delay and express costs are listed in Table 7.5. There are three possible priority levels; low, normal and high. Furthermore, priority differ in penalty according to the type of facility that requests the order. Production facilities are given higher priority than drilling rigs, while pickups are given the lowest priority regardless of facility type.

Table 7.5: *Indirect costs per order. Delays are given in USD per hour overdue delivery.*

Priority	High	Normal	Low
Delivery delay, production facilities	\$ 1,000,000	\$ 100,000	\$ 40,000
Delivery delay, drilling rigs	\$ 150,000	\$ 65,000	-
Pickup delay	\$ 100,000	\$ 20,000	-
Express delivery	\$ 2,000,000	\$ 2,000,000	\$ 2,000,000

Vessel fleet

There are two types of platform supply vessels available, the PSV 1500 and the PSV 3000. Their common operating speed is 10 knots (18.5 km/h). The vessels differ mainly in terms of deck capacity and costs. The deck spaces are 240 m^2 and 620 m^2 for the PSV 1500 and the PSV 3000, respectively. As mentioned, each vessel is normally utilized for two voyages a week. Assuming 12 hours for each loading operation sets the total sailing limit to 3 days (72 hours) per voyage. It is assumed no binding sailing distance limit for any vessel in Campos basin. Furthermore, refueling offshore requires 30 minutes of docking and undocking at the refueling tanker, in addition to approximately 3 hours required for the refueling operation. Hence, a total of 4 hours is assumed for each refueling operation.

Order specifics

Orders range from an area requirement of 30 m^2 to 150 m^2. Additionally, the time windows for delivery and pickup orders range from 10 to 20 hours. Pickup and delivery orders related to the same platform usually have overlapping time windows, as would be expected in a realistic scenario. There is also a maximum delay of 24 hours for any order. Service times offshore range from 1 to 5 hours, depending on the size of the order. Delay costs for overdue delivery of an order was shown in Table 7.5.

Scenarios

Based on the mentioned parameters, seven realistic scenarios have been constructed from Campos basin. The various scenarios differ in terms of the length of the planning horizon, number of facilities, number of buoys, vessel fleet and number of orders. Their key parameters are listed in Table 7.6.

Table 7.6: *Key parameters for the Campos basin scenarios*

	CB1	CB2	CB3	CB4	CB5	CB6	CB7
Facilities	6	8	10	10	12	14	14
Buoys	2	3	3	4	4	6	4
Orders	15	20	25	30	30	35	40
Departures	4	4	5	5	6	7	7
Refueling tankers	1	2	2	2	2	4	2
PSV 1500	3	3	3	3	4	5	5
PSV 3000	3	3	3	3	4	5	5

The facility positions are chosen randomly from Figure 7.2. The number of orders is somewhat proportional to the number of facilities and departures to ensure a match between capacity and the PSV fleet available. The PSV fleet is set such that both vessel types must be utilized during the planning horizon. The fleet of tankers is chosen such that there are more available buoys than tankers in each scenario. The buoys available are chosen in close proximity of the majority of facilities.

Chapter 8

Computational Study

The motivation of this chapter is to perform both technical and economical analyses on the arc-flow and path-flow models, based on the instances from Campos basin presented in Chapter 7. We have argued that an important property of an operational model is its ability to find good solutions quickly. Therefore, in Section 8.1 we conduct a technical study on the established path-flow model to find the best approach to shorten the solution times. In Section 8.2 the path-flow model's performance is compared to the performance of the arc-flow model. Then, in Section 8.3, the models are utilized on one of the Campos basin scenarios by varying PSV fleet size and mix, refueling fleet and penalty costs to evaluate these input parameters' influence on total costs. Lastly, in Section 8.4 we discuss some direct challenges for Petrobras related to the implementation of the operational model in practice.

8.1 Technical aspects of the path-flow model

The purpose of this section is to test various technical aspects of the path-flow model to establish appropriate settings for an efficient program. Three aspects are considered and evaluated with the scenarios from Campos; column generation procedures, frequency of the IP heuristic and utilization of parallel computing. The areas covered are only a selection of technical aspects that could be considered. However, the topics have been chosen consciously based on previous discussion, literature study and additional testing.

8.1.1 Column generation

The procedure of adding columns to the restricted master problem is an important feature of a Branch and Price algorithm. We have considered three procedures that were discussed in Chapter 6. The first procedure is called break-first-negative (BFN), which stops the search for new solutions in the subproblems as soon as a column with negative reduced cost has been found. This column is then added to the RMP, and a signal is given that this subproblem has finished. BFN will not create a lower bound on the optimal objective value before the root node has been solved to optimality in the RMP. In addition, each node in the B&P search tree must be solved to optimality before it can be pruned.

The second procedure is called solve-sub-optimum (SSO), which solves all subproblems to optimality in each iteration of the DW algorithm. The optimal column is then returned to the RMP. This procedure creates a lower bound on the optimal objective value in the root node of the B&P tree according to equation (7.7). Furthermore, using this equation to create a lower bound on subsequent nodes in the B&P tree might lead to pruning before the node has been solved to optimality.

A third procedure is also considered. In this case a combination of BFN and SSO is implemented. This procedure, named break-first-then-optimum (BFO), is identical to the BFN for the n first iterations of the DW algorithm in the root node. At iteration $n+1$ the procedure switches to SSO for the remainder of the program.

The three procedures are run for five case instances; CB1-CB5. However, we base most of the discussion on CB2 and CB5 for simplicity. Table 8.1 shows outputs from running the model for these two problem instances.

Table 8.1: *Comparison of column generation procedures for CB2 and CB5*

	CB2			CB5		
	BFN	SSO	BFO	BFN	SSO	BFO
Time to root [sec]	117	118	96	2,324	11,917	936
Gap at root	2.43%	4.66%	2.43%	1.34%	1.49%	2.39%
Iterations to root	28	23	25	59	30	54
Number of columns	200	155	196	610	395	634
Nodes in B&P tree	7	7	7	13	13	13
Total time [sec]	**274**	**236**	**245**	**5,565**	**13,969**	**3,205**
Optimal solution found [sec]	211	195	204	5,271	13,857	2,734

For the smallest problem instance, CB2, we notice that BFN and SSO require almost the same time to solve the root node, while BFO is slightly better. Furthermore, SSO generates fewer columns, which is expected given that it provides better columns for each subproblem.

Regarding solution time, we see that SSO provided a faster convergence in the CB2 case. The program ran for 236 seconds, while the optimal solution was found after 195 seconds. In this case BFN required the most CPU time, i.e. a total of 274 seconds to prove optimum. This is probably a result of poor columns included in the RMP during the column generation. An interesting notion is that although SSO proves optimum faster, it also has the largest gap at the root node. For BFO we see that besides solving the root node faster than the other procedures, it also has a tight gap. Overall there is little distinction between the procedures for the CB2 case, in fact only 16.3% separates the longest solution time from the shortest.

For the larger problem instance, CB5, there are more dominant variations. Firstly, SSO is the slowest method in this case. This is much due to the fact that it spends about 200 minutes to solve the root node, while the fastest instance, BFO, requires roughly 16 minutes. Overall, BFO is the superior procedure for CB5, and proves optimum within approximately 54 minutes. In contrast BFN deviates by roughly 74% and SSO by 336% in total solution time. It is clear that all three procedures have small gaps at the root node, for which BFN and SSO are below 2%. This proves that despite requiring long time to prove optimum, the model has a decent solution even before the Branch and Price scheme is initiated. For instance, in certain applications it might be sufficient to terminate the BFN procedure in the root node, given a gap of 1.34%. The results illustrate how the tail-off effect commonly causes slow convergence during the B&P scheme. The problem with the SSO procedure is the time spent on the first iteration. For this instance SSO requires roughly 170 minutes to find the first column, which is a significant weakness for this procedure. Consequently, SSO requires less columns to be generated.

More surprisingly is that BFO generates more columns than BFN. A possible reason could be that poor columns are generated after the switch to SSO. Furthermore, these columns might in fact be a misguidance for the solution process, and thus require more columns to be generated. It would make more sense if BFO generated less columns than BFN, as it includes the SSO procedure, which clearly results in fewer columns. However, these results show that the number of columns generated for each procedure is highly problem dependent.

Overall it seems that as the problem size increases, the BFO procedure becomes superior. In both instances, we see that BFO solves the root node faster than the other two procedures. Once the problem size increases, there is a difficulty related to generating initial columns using SSO. But we also know that for the BFO procedure, that the first few iterations are run with BFN, meaning that most of the search is done using SSO even if the procedure stand-alone is the weakest. As soon as we have a decent pool of columns, the BFN procedure seems to experience slow convergence, tail-off effect, and SSO is necessary to speed up the process.

Figure 8.1 provides an overview of the total solution times for the three procedures based on five cases (CB1-CB5). The figure illustrates that BFO causes faster convergence than the stand-alone procedures in all instances. It is however difficult to give an exact comparison between BFN and SSO. For CB1–CB4 we see that SSO performs better, while for CB5 it is apparently less convenient to use SSO. Based on these results we have no basis to draw a definite conclusion. This is another indication that the solution process of the path-flow model is highly problem dependent.

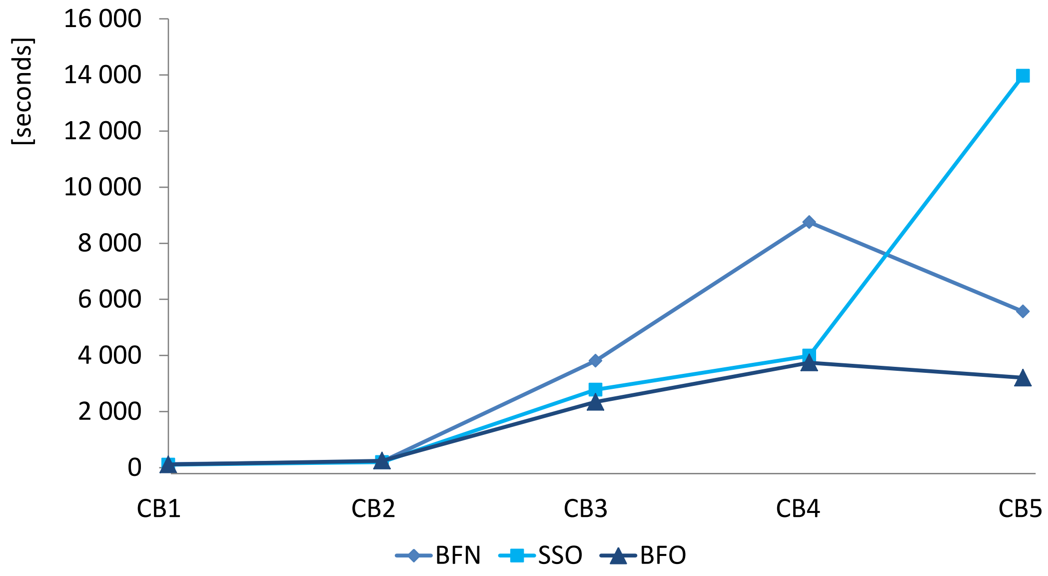

Figure 8.1: *Total solution times for CB1-CB5*

In Figure 8.2 we compare upper bounds for the three procedures until the root node has been solved. For evaluation purposes a medium-sized problem instance is used, in this case represented by CB3.

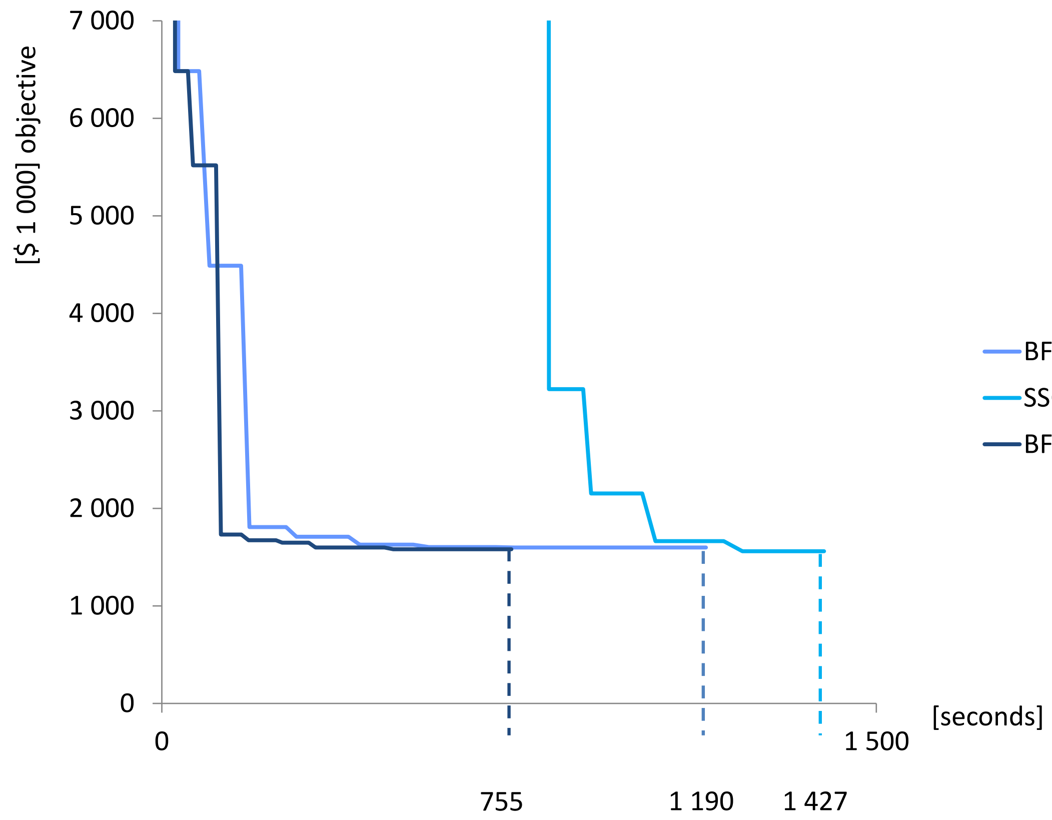

Figure 8.2: *Upper bounds until the root node using the three column generation procedures, given by the CB3 test case. Dashed line indicates that the root node has been solved*

The figure shows that BFO solves the root node first, followed by BFN and SSO respectively. These results are consistent with the results from Table 8.1, for CB2 and CB5. We see that BFN and BFO have almost equivalent bounds initially during the solution process, which would be expected. The faster convergence for BFO is mostly due to the SSO switch after a few iterations. For the stand-alone SSO procedure, the problem is that the model requires a substantial amount of time to find the first optimal column. However, the graph also illustrates that once the model moves past the first iteration, the convergence is rather efficient.

Based on the results shown so far, it is evident that a mixed column generation scheme will most likely be the most efficient approach. An interesting aspect of this procedure is how many iterations (n) that should be run with the BFN approach before the switch to SSO is conducted. In Figure 8.3 the BFO procedure is tested by assigning varying values on the number n based on CB4 and CB5. Note that for the previous results shown in this section we have used a frequency count of four, i.e. $n = 4$. The major axis in the figure provides the total time to solve the root node.

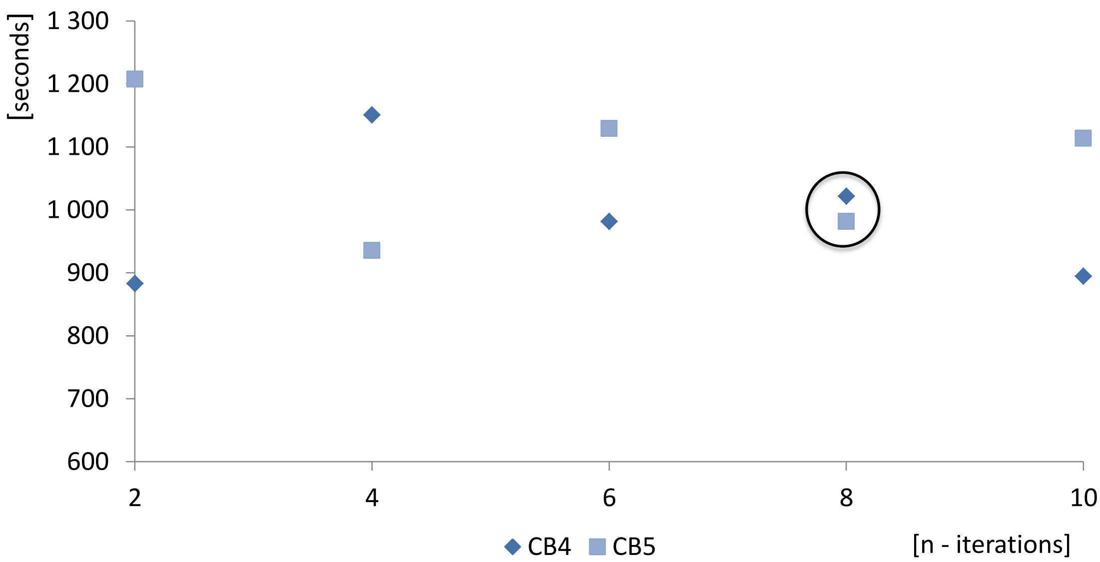

Figure 8.3: *Time to root node using the BFO procedure with different number of iterations n for CB4 and CB5*

Based on the plots in the figure there is no definite intuition regarding appropriate number of iterations for the BFO procedure. Yet again, this aspect seems to be problem dependent. For the purpose of the continuation of the study, we have set the number of iterations to eight. This selection is circled out in Figure 8.3, and is chosen due to the consistency of the two instances and the low average solution time.

8.1.2 Frequency of the IP heuristic

In this section we observe how the solution time is affected the frequency of the IP heuristic. Solving the RMP as an IP might be computationally intensive, but it could also help establish better upper bounds for the optimal objective value. Two problem instances are tested. The results from running CB2 and CB5 are illustrated in Table 8.2 and Table 8.3, respectively.

For each case, the problem is tested using six different frequency settings for the IP heuristic. The frequency row in the tables represent how frequent the heuristic is run in the program, e.g. a frequency of 2 means that in every other iteration the program solves the RMP-IP with the current pool of columns. The frequency setting also applies to the B&P tree, however in this case it means that the heuristic is run for every other node. Thus, it runs at most once in any given node, after the RMP-LP is solved to optimum. The frequency setting of 100 implies that the heuristic is run only in the root node. The results also show how the optimal solution is found, i.e. whether it is found by applying the IP heuristic or if the branching scheme provides an integral solution.

In Table 8.2 we see the results when running the CB2 instance. The first observation is that for different frequency settings we get some deviations in the number of iterations to solve the root node. This aspect shows an important notion about the solver application. In theory, these settings should not affect the iteration count to the root, because the IP heuristic is unrelated to processes of finding the optimal RMP-LP. Apparently, there are some built-in functions in Xpress that may lead to certain variations in the solution output. Thus, we need to be aware that the results are not perfectly comparable for all instances. However, the instances where the results have the same iteration count should experience rather similar solution processes.

Table 8.2: *Alternative IP heuristic frequencies for CB2*

	CB2					
Frequency	**1**	**2**	**3**	**4**	**5**	**100**
Iterations to root	25	28	25	25	28	27
Time to root [sec]	100	113	96	97	115	106
Optimal found by	Branch	Heuristic	Heuristic	Heuristic	Branch	Branch
Optimal found [sec]	173	91	73	87	214	183
Total time [sec]	248	269	245	245	276	245

In Table 8.2 we see that the instances with frequencies 1, 3 and 4 have the same number of iterations to reach the root. In Table 8.3 the same applies for the frequency settings 1, 2 and 100.

Table 8.3: *Alternative IP heuristic frequencies for CB5*

	CB5					
Frequency	**1**	**2**	**3**	**4**	**5**	**100**
Iterations to root	63	63	52	54	52	63
Time to root [sec]	1,125	1,116	915	949	926	1,116
Optimal found by	Branch	Branch	Branch	Branch	Branch	Branch
Optimal found [sec]	3,086	3,096	2,698	2,766	2,289	3,104
Total time [sec]	3,611	3,600	3,146	3,240	2,894	3,608

Given the frequencies 1, 3 and 4 for CB2 we notice that there is a very small difference in time to the root node. Given that the IP heuristic is the only aspect separating the instances, we would expect longer solution times for the more frequent settings. This is indeed true when looking at frequency 1. However, for instance 3 and 4, we see that frequency 4 is actually slightly more time consuming.

For frequency settings 3 and 5, which require the same amount of iterations to solve the root node in the CB5 instance, the same notion is observed where frequency 5 requires more time to the root node, despite running the heuristic less frequent. Looking at the remaining frequency instances, i.e. 1, 2 and 100, there is a certain level of correlation indicating that the number of times the heuristic is run may cause a delay to reach the root node. The differences are however very small and could be caused by other uncontrollable factors.

Note that for both CB2 and CB5 the IP heuristic does not influence the number of nodes in the B&P tree. This implies that all nodes have to be searched in all instances, regardless of how early the optimal solution is found. The IP heuristic does however manage to find the optimal solution before the root node is solved in CB2 for frequencies 3, 4 and 5, yet rather surprisingly not for frequency 1. Although the heuristic does not reduce the tail-off effect of the B&P tree, it manages to create good solutions rather quickly. For CB5 we see that the optimal solution was found by branching, regardless of IP heuristic frequency.

Overall, what these results show is that running the heuristic frequently may inflict additional time consumption, but compared to other aspects of the program it is almost negligible. It is apparent that the subproblems constitute most of the time consumption in the program, while the RMP is solved fairly quickly. Lowering the upper bound as we go provides the planner with good solutions. As the problem size increases, this is valuable if the MILP gap has not been closed upon termination of the algorithm. Based on the results and the notion concerned with the low processing time related to the IP heuristic, we may conclude that the heuristic should be set at a frequency level of 1 for the model in question.

8.1.3 Sequential versus parallel procedure

The potential of parallel solving of subproblems in the path-flow model was discussed in Chapter 7. Barrier synchronization is the most simple form of parallel computing, in which the RMP waits for all SPs to finish before any further executions are made. The path-flow model has been implemented both sequentially and with the use of the barrier synchronization principle. However, due to communication disruptions on the cluster, what was sought after was not achieved. Instead, when subproblems were solved in parallel, the outcome turned out slower than with an equivalent sequential algorithm. How Xpress-MP communicates between files on the cluster is out of scope of this thesis, therefore only a theoretical gain of utilizing parallel computing will be presented here as an interesting area of extension to reduce CPU time. Figure 8.4 illustrates by a general example how parallel computing uses less time per iteration compared to a sequential procedure.

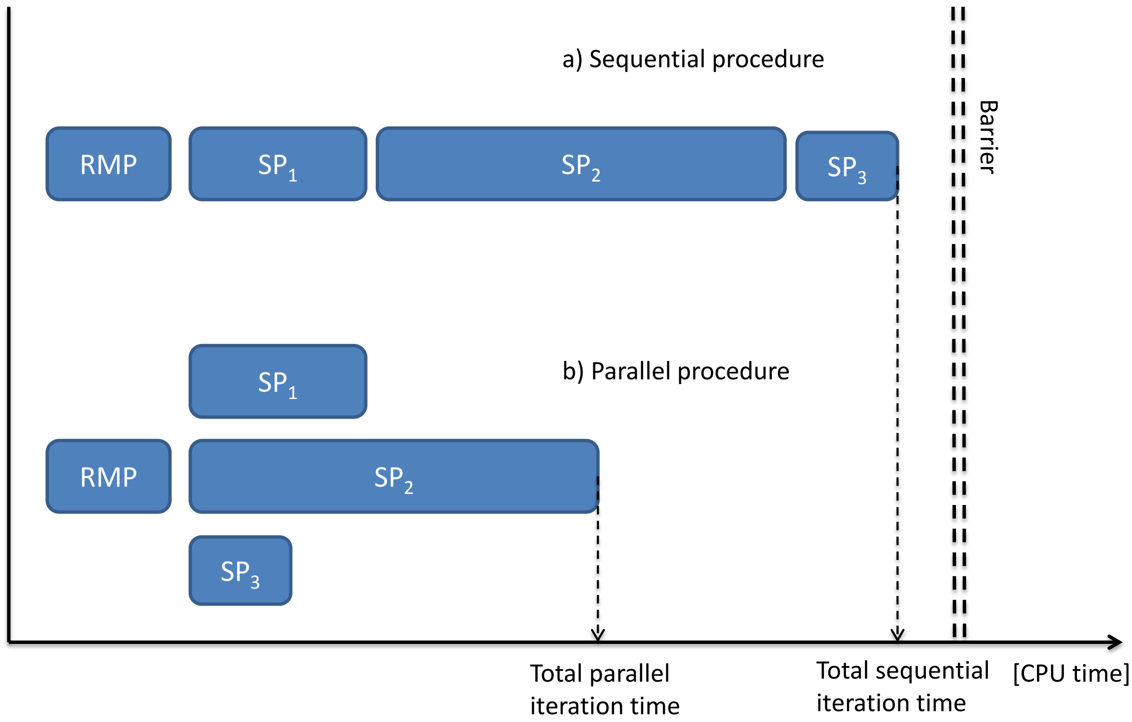

Figure 8.4: *Comparison of processing time for parallel and sequential procedure*

Table 8.4 shows the computational times required to solve each subproblem in the CB2 case for the first five iterations of the DW algorithm, using a BFN column generation procedure. The time required to solve the RMP using the initial columns has not been regarded.

Table 8.4: *Computational times for sequential solving of the CB2 case [seconds]*

Subproblem	V1D1	V1D2	V1D3	V1D4	V2D1	V2D2	V2D3	V2D4	SPs only	Total time	Theoretical best
Iteration 1	0.13	0.83	1.16	1.14	0.05	0.46	0.67	1.33	5.76	5.77	1.33
Iteration 2	0.20	0.22	0.31	0.51	0.07	0.16	0.32	0.52	2.31	2.32	0.51
Iteration 3	0.09	0.15	0.23	0.27	0.13	0.35	0.25	0.35	1.83	1.83	0.35
Iteration 4	0.15	0.36	0.14	0.21	0.21	0.27	0.14	0.18	1.65	1.66	0.36
Iteration 5	0.16	0.31	0.34	0.31	0.10	0.12	0.34	0.37	2.04	2.05	0.37
Total									**13.59**	**13.63**	**2.94**

The CB2 case has a total of 4 departures and 2 vessel types, which sums up to 8 subproblems. The computational times listed in Table 8.4 are clocked in the 8 SPs, respectively. Thus, the column *V1D1* shows that subproblem corresponding to vessel 1, departure 1 was solved in 0.13 seconds in the first iteration.

The column marked *SPs only* is the summation of each individual SP's computational time, not taking into account the time required for communication between files and time required to solve the RMP. The data under the column marked *Total time* is clocked in the RMP file, and therefore includes solving all subproblems, communication between files and the time required to solve the RMP in each iteration. The *Theoretical best* states the time required to solve the slowest subproblem for a given iteration, e.g. for iteration 1 subproblem *V2D4* is the slowest.

We can see that the time required for communication and solving the RMP is almost negligible compared to the time required for solving SPs. This suggests a great potential for parallel computing. Using simple barrier synchronization, the total theoretical best for solving only the SPs, during the first 5 iterations of the DW algorithm, is 2.94 seconds. For parallel computing to function properly, the SPs must be efficiently distributed on the cores of the CPU. Since there are 12 cores of the CPU on the cluster, this particular case should theoretically split out evenly on 8 of them. A theoretical improvement factor of 4.6 should therefore be achievable from the 13.59 seconds required to solve the SPs with the sequential algorithm.

Achieving this result is however not straightforward. Firstly, as the B&P algorithm proceeds, the size of the RMP increases, and thus more time is required to solve it. On the other hand this is also true for the subproblems as their reduced costs become less negative, which may even out the effect. Secondly, as was experienced in our implemented model, communicating efficiently between files and splitting the SPs evenly on the cores of the CPU depend on the software that is being used. The module *mmjobs* in Xpress-MP that handles these tasks does unfortunately not give the desired result. For the reader's interest, the time required to do 5 iterations of column generation for CB2 with the parallel method was 20.06 seconds on the cluster.

8.2 Comparing the path-flow and arc-flow models

In this section the customized path-flow model is tested against the arc-flow model. The arc-flow model is implemented as described in Chapter 7, whereas the path-flow model is based on the findings of this chapter. The path-flow model is therefore implemented with a BFO column generation procedure, with 8 initial iterations, and an IP heuristic frequency set to 1. The subproblems are solved by a sequential algorithm.

Both models are implemented in Xpress-MP. All 7 test cases are run for a maximum of 10 hours (36 000 seconds) on the cluster. The CPU times required for closing the MILP gap for both models are shown in Table 8.5. If the gap is not closed after 10 hours, no value is reported. Furthermore, for the cases applicable, gaps for the two models are shown in Figure 8.5 after 1 hour, 3 hours and 10 hours, respectively.

Table 8.5: *Time to close duality gap for the arc-flow and path-flow models [seconds]*

	CB1	CB2	CB3	CB4	CB5	CB6	CB7
Arc-flow	7	16	-	-	-	-	-
Path-flow	130	254	3,188	5,327	4,739	8,062	-

From Table 8.5 we can see that the arc-flow model performs better than the path-flow model on the two smallest test cases. However, the limitations of the arc-flow model are evident when the size of the problem increases. The arc-flow model does not manage to close the gap within 10 hours on any instance larger than CB2. The path-flow model manages to find the optimal solution and close the gap in less than two hours for the first six test instances. For CB7, the path-flow model terminates with a gap after 10 hours of runtime.

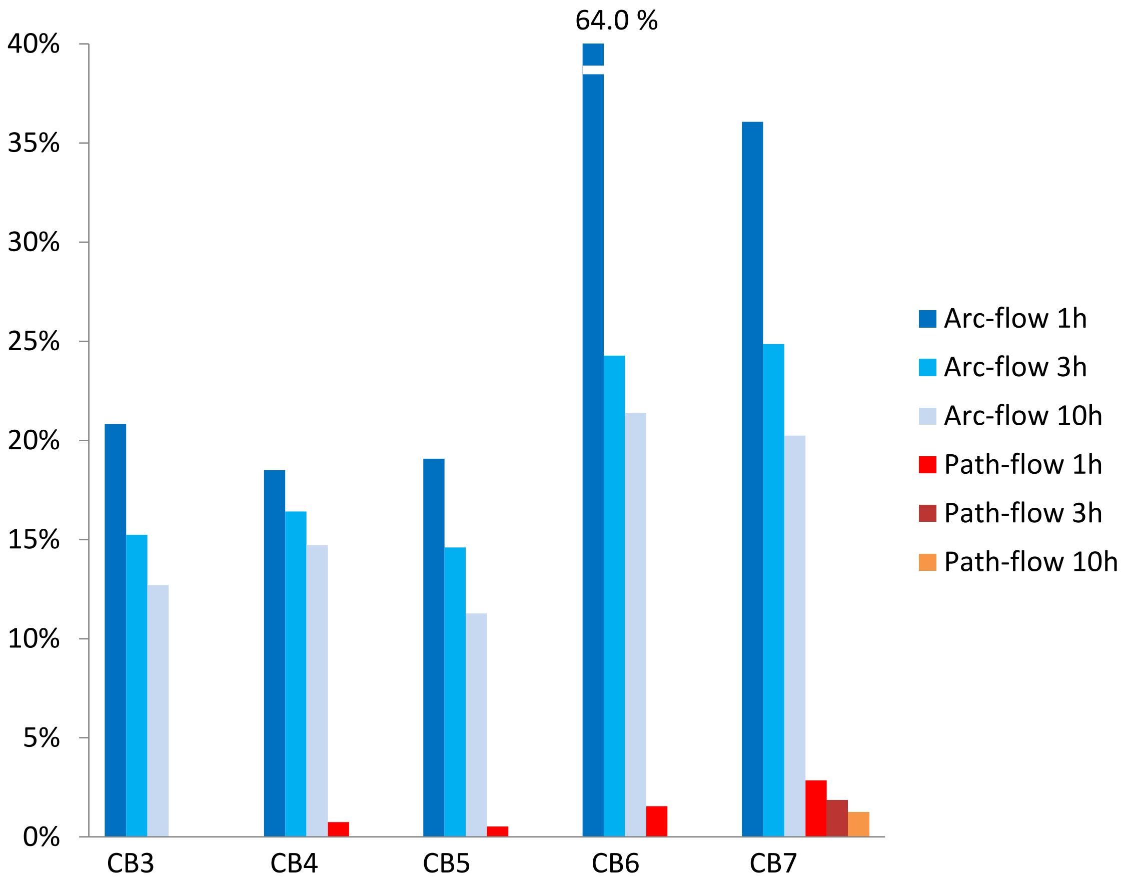

Figure 8.5: *MILP gaps after 1 hour, 3 hours and 10 hours for the arc-flow and path-flow models*

From Figure 8.5 we see that the arc-flow model experiences slow convergence. Looking at for instance the CB4 case, the gap of the arc-flow model reduces from around 18.5% after 1 hour, to 14.7% upon termination after 10 hours. The results are similar for

CB3, CB5, CB6 and CB7. It is clear that the path-flow model outperforms the arc-flow model significantly on larger problem instances, as it manages to reduce the MILP gap quickly. After one hour, the gaps are below 3% for all test instances in the path-flow model.

The difference in MILP gaps indicates the path-flow model's superiority. However, this gap is created by an upper pessimistic bound and a lower optimistic bound, as discussed in Chapter 6. Although there exists a MILP gap upon termination of the arc-flow model, the upper bound on the objective value, thus its currently best IP solution, might still be close to the optimal one. We would therefore like to investigate whether the arc-flow model finds good solutions early in the process.

Table 8.6: *Best IP solutions for the arc-flow and path-flow models after 3 hours*

	CB3	CB4	CB5	CB6	CB7
Arc-flow IP	1,560,443[a]	1,548,003	1,606,757	1,890,560	1,890,425
Path-flow IP	1,560,443[a]	1,503,663[a]	1,568,677[a]	1,717,940[a]	1,624,000
Difference	0.0%	2.9%	2.4%	10.0%	16.4%

[a] Optimal solution

Table 8.6 shows the best IP solutions for the arc-flow and path-flow models after 3 hours, for CB3-CB7. Note that the path-flow model has found the optimal solution for all but CB7. For the CB3 case, the arc-flow model has actually found the optimal solution, and the entire duality gap in Figure 8.5 is created due to the lower bound. For CB4 and CB5, the arc-flow model is only 2.9% and 2.4% from the optimal solution, with MILP gaps of 16.4% and 14.6%, respectively. This means that the lower bound is the main contributor to the gap, and the pessimistic bound is rather low after 3 hours. For CB6 and CB7 on the other hand, the differences from the best path-flow objective are 10.0% and 16.4%, respectively. Their respective duality gaps in Figure 8.5 are 24.3% and 24.9%. This gives an indication that the pessimistic bounds on the arc-flow model are rather large for these instances after 3 hours.

We see that the arc-flow model is able to find good solutions for some of the instances after 3 hours, and computational time is most spent on raising the lower bound by discarding optimistic nodes in the B&B tree. For CB6 and CB7, the difference from the optimal solution after 3 hours is however significant. Overall, the test instances therefore find little evidence that good IP solutions are found quickly in the arc-flow model. This confirms the superiority of the path-flow model for the larger problem instances.

An interesting aspect regarding the path-flow model can be seen from the gap of CB7 in Figure 8.5. The gap after 1 hour is only 2.9%. After 3 hours, this gap is reduced to 1.9%. After another 7 hours, when the model terminates, the gap is still at 1.3%. This shows that the DW algorithm of the path-flow model is highly efficient in reducing the optimality gap quickly in the root node. The DW algorithm manages to create good solutions in a relatively short amount of time. However, removing this gap entirely is computationally costly with the B&P algorithm, due to the tail-off effect.

When the number of variables to branch on in the subproblems becomes large, the number of possible branches in the B&P tree also increases, and the search becomes rather inefficient. This shows one limitation and a possible area of improvement for the path-flow model.

8.3 Economical study

The intention of this section is to apply the implemented path-flow model to a base case in order to evaluate the model from an economical perspective. For the base case (BC) we have chosen the third problem instance of Campos basin (CB3), shown in Table 7.6. The study is divided into four parts, for which we evaluate how various alterations affect the performance of the model. Figure 8.6 gives an overview of the main areas covered in this section. Each area of comparison is dedicated a numbered subsection.

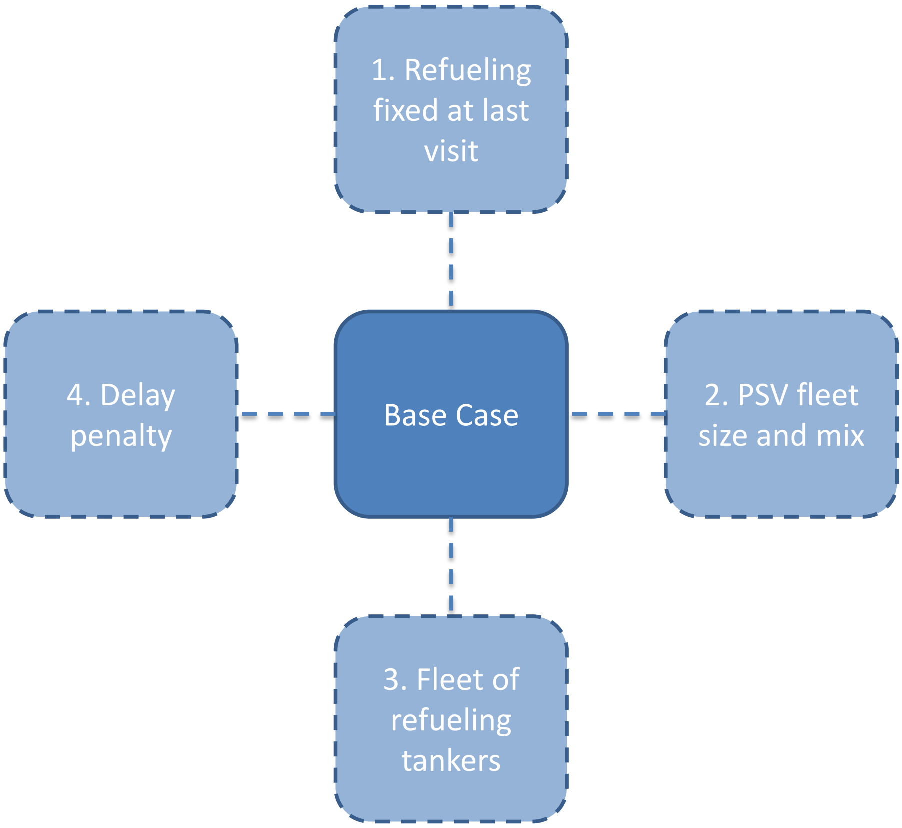

Figure 8.6: *Areas of comparison for the economical study*

In the previous section the main objective was to make improvements in the context of making an efficient operational model, i.e. reduce the CPU time. In terms of economical value, knowing how the model is affected by the input parameters could add additional value on a strategic level. Although the overall CPU time is presented along with the results, it will not be the primary focus of this study. Even so, it could give an interesting intuition regarding the sensitivity of solution times with respect to input alterations. Note that the cases are run on a desktop computer, therefore the solution times might vary from the results in the previous sections.

Base case

Figure 8.7 shows the most important inputs for the base case. Ten facilities are chosen from the original positions shown in Figure 7.2. A set of 25 orders (both pickup and delivery) are divided between the platforms. Furthermore, two refueling tankers are available for deployment at two out of three buoy positions. The planning horizon is five days, meaning that five departures are considered. As usual, two vessel types are available, PSV 1500 and PSV 3000. The fleet size and mix are chosen consciously in order to ensure that both vessels are utilized.

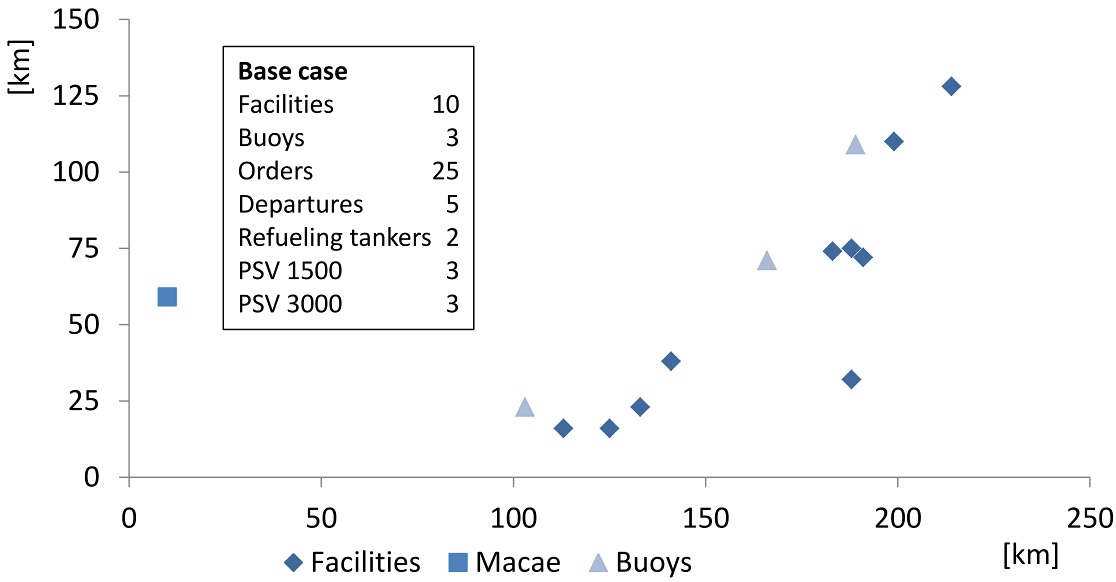

Figure 8.7: *Placement of nodes in the base case*

The figure also shows the relative placement of all nodes included in the base case, i.e. the offshore production facilities, depot and buoy positions. The nodes are deliberately chosen with a rather wide spread for evaluation purposes.

Table 8.7 gives the results from running the base case. The direct costs included in the first row correspond to the variable sailing costs and fixed costs of leasing the respective vessel chosen for the departures. Furthermore, the number of delayed orders and express orders are included. Total delay is the summation of delayed hours for all orders, which is 13 hours in this case.

Table 8.7: *Results from the base case*

	Base Case
Direct costs	$ 1,257,200
Percentage from BC	**0.0%**
Delayed orders	3
Express orders	0
Total delay [hours]	13.0
Indirect costs	$ 303,243
Percentage from BC	0.0%
PSV 1500	2
PSV 3000	3
Time to close MILP gap [sec]	1,511
Percentage from BC	0.0%

The indirect costs correspond to the summation of delay costs and express delivery costs. As opposed to direct costs, these costs are not true values, but rather symbolic factors used for modeling purposes. The indirect costs are affected by the predetermined cost of delay, which will be discussed later. The base case output shows that two and three vessels are chosen of PSV 1500 and PSV 3000, respectively. All the results are included in the remaining tables as a benchmark for analysis.

8.3.1 Refueling fixed at last visit

In the base case there are no restrictions with regard timing of refueling operations. This means that the sequence in which a PSV visits a tanker is not binding. The intention of this analysis is to simulate the current situation for PSVs in Campos, in which case the they visit a refueling tanker as their last stop before returning to port. The results from this problem instance is shown in Table 8.8. The results state that, besides a small increase in direct costs (0.4%), the solutions are similar. It is important to point out that even if the fixation made by this alteration makes the problem more restricted, the positioning of the refueling tankers still lead to good solutions. In this case, the sequence in which refueling tankers are visited does not influence PSVs ability to deliver orders on time. Based on the positioning of buoys, it is no surprise that the solutions are almost equivalent. Figure 8.7 shows that most of the buoys are located in close vicinity of the platforms, in the space between the depot and the platforms. Thus, visiting a tanker at the end of the voyage may in fact be optimal in terms of direct costs.

Table 8.8: *Deviations when refueling is fixed at last visit*

	Base case	Refuel last
Direct costs	$ 1,257,200	$ 1,262,520
Percentage from BC	**0.0%**	**0.4%**
Delayed orders	3	3
Express orders	0	0
Total delay [hours]	13	13
Indirect costs	$ 303,243	$ 303,243
Percentage from BC	0.0%	0.0%
PSV 1500	2	2
PSV 3000	3	3
Time to close MILP gap [sec]	1,511	492
Percentage from BC	0.0%	-67.4%

We also see that the solution time is rather surprisingly reduced by 67.4%. Less route combinations are possible when we remove all arcs going from the buoys to the platforms, which is one possible explanation of the sudden drop in solution time.

8.3.2 PSV fleet size and mix

An important strategic decision in any routing problem is deciding the appropriate fleet size and mix. This aspect has been evaluated by applying six alterations to the model input related to the size and mix of the PSV fleet, shown in Table 8.9. Note that we have omitted removing only one PSV 1500 from the initial fleet of three, since the solution found would be identical to the base case solution (for which only two out of three PSV 1500 are utilized).

Firstly, an additional PSV 3000 is added to the mix. These results are shown by the second column from the left in Table 8.9. In this case the results remain unchanged. The base case solution chooses a mix of three PSV 3000 and two PSV 1500, respectively. These results prove that the optimal mix for the CB3 case is in fact the results from the base case.

When removing a PSV 3000, and thus reducing the fleet size, the problem becomes more constrained. Naturally, the model compensates by allocating an additional PSV 1500. Despite less overall capacity, the results show that the model solves the problem with the same delay count. In terms of fixed and variable costs there is a trade-off related to this alteration. The fixed costs are reduced due to a less expensive vessel, but the routes are also less efficient due to less capacity on each vessel. Hence, the variable sailing costs are higher for this instance. The result is that the trade-off in costs is almost completely offset by this alteration. This may further explain why

Table 8.9: *Deviations in PSV fleet size and mix*

	Base case	**+1 PSV 3000**	**-1 PSV 3000**	**-2 PSV 3000**	**Only PSV 1500**	**-2 PSV 1500**	**Only PSV 3000**
Direct costs	$ 1,257,200	$ 1,257,200	$ 1,261,400	$ 1,151,500	$ 1,049,860	$ 1,334,100	$ 1,419,740
Percentage from BC	**0.0%**	**0.0%**	**0.3%**	**-8.4%**	**-16.5%**	**6.1%**	**12.9%**
Delayed orders	3	3	3	1	3	3	3
Express orders	0	0	0	1	4	0	0
Total delay [hours]	13.0	13.0	13.0	1.0	27.0	13.0	13.0
Indirect costs	$ 303,243	$ 303,243	$ 303,243	$ 2,063,243	$ 9,488,243	$ 303,243	$ 303,243
Percentage from BC	0.0%	0.0%	0.0%	580.4%	3,028.9%	0.0%	0.0%
PSV 1500	2	2	3	4	5	1	0
PSV 3000	3	3	2	1	0	4	5
Time to close MILP gap [sec]	1,511	1,608	963	669	9,194	257	753
Percentage from BC	0.0%	6.4%	-36.3%	-55.7%	508.5%	-83.0%	-50.1%

the solution time is higher for the base case. Since the trade-off between the vessels result in almost equivalent solutions, it could imply that the model is left with many promising scenarios to consider, which could further increase the CPU time.

In the next instance, two PSV 3000 vessels are removed from the mix and replaced by additional PSV 1500 vessels. Due to the decrease in capacity, the vessels are no longer capable of carrying out all orders. An express vessel is necessary, which results in a significant increase in indirect costs. However, we also notice that the direct costs are reduced by 8.4%. This is related to the fact that we have a cheaper fleet of vessels, which leads to a reduction in both fixed and variable sailing costs. Another interesting observation from this instance is that the overall delay costs are reduced significantly by assigning one order to an express vessel. In other words, we see that the order corresponding to the express delivery does in fact put a lot of strain on the problem. The model formulation in this thesis will never deliberately choose an express delivery, however, in real life this case may constitute a situation where it would be beneficial to do so.

For the instance with only PSV 1500 the results show a total of four express departures. Clearly, the direct costs would decrease since less orders are serviced on regular departures, leading to a decrease in variable costs. Fixed costs are also reduced due to a cheaper vessel fleet. However, the case described here shows a situation where a sub-optimal fleet mix leads to capacity problems and thereby cause an increase in indirect costs. Note also that solution time is increased significantly due to this restriction.

In the last two columns, the number of PSV 1500 vessels are reduced, forcing the model to choose the larger vessel. These cases describe situations where there is excess capacity. We know that both the variable and fixed sailing costs are higher for the PSV 3000, which lead to an increase in overall direct costs of 6.1% and 12.9%, respectively. The results show that wrongful determination of fleet size could lead to significant costs related to excess capacity. It is worth noting that for the current PSV fleet in Petrobras Logistics, a majority of the vessels is in fact PSV 3000.

8.3.3 Fleet of refueling tankers

In relation to the allocation of refueling tankers we evaluate two alterations. The first instance describes a situation where the tanker positions are not optimized, but instead fixed at random positions. The next case describes a situation where the fleet of refueling tankers are decreased by one, i.e. only a single tanker will be allocated to a buoy.

The first instance in Table 8.10 shows only a small increase in sailing costs as a result of the random deployment. As for the other parameter values, the solution remains unchanged. For the second instance, the results have a similar interpretation. However, in this case there would most likely be much strain on the single tanker. With regard to the assumptions made in the formulation, the model does not account for all aspects. For instance, PSV oilers that also make use of the tankers limit

Table 8.10: *Deviations in the fleet of refueling tankers*

	Base case	Fixed tanker position	-1 tanker
Direct costs	$ 1,257,200	$ 1,265,600	$ 1,271,200
Percentage from BC	**0.0%**	**0.7%**	**1.1%**
Delayed orders	3	3	3
Express orders	0	0	0
Total delay [hours]	13	13	13
Indirect costs	$ 303,243	$ 303,243	$ 303,243
Percentage from BC	0.0%	0.0%	0.0%
PSV 1500	2	2	2
PSV 3000	3	3	3
Time to close MILP gap [sec]	1,511	1,167	1,701
Percentage from BC	0.0%	-22.7%	12.6%

the number of vessels that may refuel simultaneously. Thus, in practice there are other considerations that could be affected by the reduction of refueling tankers. Overall, both instances indicate that the buoys are well positioned, and changing the deployment location makes little difference in terms of direct costs.

8.3.4 Delay penalty

Table 8.11 presents four new cases where we have altered the delay penalty factor. For the first two instances the penalty is decreased. The last two instances show the results when the penalty is increased by a certain factor. The motivation for doing this analysis is to observe the trade-off between minimizing sailing distance and minimizing delay.

Table 8.11: *Deviations in the delay penalty factor*

	Base case	Half penalty	No penalty	Penalty x 2	Penalty x 5
Direct costs	$ 1,257,200	$ 1,242,080	$ 1,119,300	$ 1,304,660	$ 1,316,560
Percentage from BC	**0.0%**	**-1.2%**	**-11.0%**	**3.8%**	**4.7%**
Delayed orders	3	2	5	3	2
Express orders	0	0	0	0	0
Total delay [hours]	13	14.1	40.5	10.9	7.7
Indirect costs	$ 303,243	*$ 163,243*	*$ 0*	*$ 555,676*	*$ 1,375,000*
Percentage from BC	0.0%	*-46.2%*	*-100%*	*83.2%*	*353.4%*
PSV 1500	2	2	3	3	2
PSV 3000	3	3	2	2	3
Time to close MILP gap [sec]	1,511	1,578	2,960	1,530	2,077
Percentage from BC	0.0%	4.5%	95.9%	1.3%	37.5%

When the delay penalty is halved, there is a slight reduction in direct costs. Although the number of delayed orders are reduced by one, the number delayed hours has increased. Naturally, the indirect costs are not representative in this case, because the cost of delay is different for the two instances.

For the next case the delay penalty is removed entirely. Based on the results from this test run we are able to make some interesting observations. Firstly, the direct costs are reduced by 11%. Since the model no longer is penalized for delays, the main objective is to optimize the sailing route based on distance. Thus, the direct costs are minimized. We also see that the solution time is almost doubled when direct costs is the only factor of the objective function. Furthermore, the total delay is 40.1 hours compared with 13 hours in the base case. The case also shows that the optimal PSV mix is changed, where PSV 1500 is the preferred choice for most of the departures. Overall, it is clear that the downside of this approach outweighs the improvements of the sailing route. In fact, it may be argued that reduction in direct costs is actually very small given the different focus areas in the two cases. That is, the base case provides good routes even though the emphasis is to minimize penalty violations.

This theory can be further strengthened by the last two instances where penalty is increased. The results show the opposite effect, i.e. the direct costs increase while the hours of delay are reduced. As implied earlier, these alterations only result in a minor increase in direct costs; 3.8% and 4.7%, respectively. Again, it is apparent that the model provides good routes despite higher penalties for delays. In terms of total delay, we see that there is a significant reduction from the base case. When twice the penalty is applied we observe a reduction of 2.1 hours, while increasing the penalty costs by a factor of 5 results in a reduction of 5.3 hours.

These results sum up an important intuition regarding the sensitivity of the delay penalty. In terms of minimizing the sailing distance and providing optimal routes, the exact value of the penalty may in fact not be critical. The results show that deviations in direct costs are small even if the focus becomes more centered on minimizing delay, and it shows similar results when sailing costs are emphasized. This conclusion has a natural interpretation which may be observed in Figure 8.8. The figure illustrates two basic incidents where a) describes the scenario where delay penalty is enforced, and b) shows the optimal route with respect to distance (i.e. the *No penalty* column in Table 8.11). In the figure we observe that the nodes, which simulate platforms, are located fairly close to each other. Another observation is that the distance between the depot and the cluster of platforms by far represents the most dominant range. This begs to show that the internal routing between the platforms in many instances may in fact have a rather small significance with respect to overall costs. Hence, based on these results, the main objective should still be to minimize delay costs. With regard to sailing costs, the analysis shows that the routes are likely to be good, even with a secondary focus on direct costs.

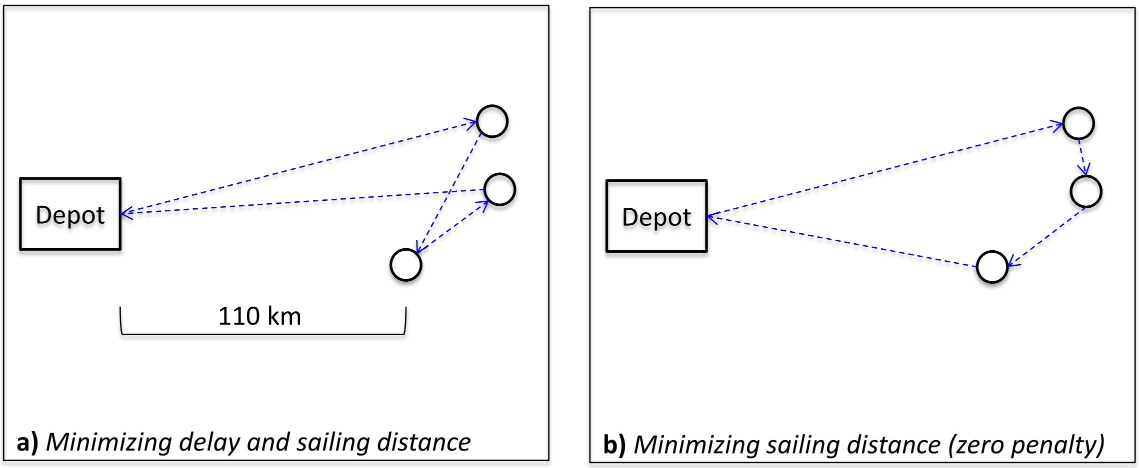

Figure 8.8: *Changes in sailing distance based on factor of delay penalty*

8.4 The model in practice

This chapter has considered some technical and economical aspects of the PSVRSP-RT model. In this section we will discuss how the model fits in a real planning environment, and how the model may add value as a decision support tool for Petrobras in the future.

The problem has been modeled by making some assumptions of the real world. Simplifications are made on factors such as weather's influence on sailing speed and other parameters that are assumed known for the entire planning period. In practice these parameters are in fact highly uncertain. The uncertain factors can be split into external and internal uncertainty (Wallace, 2003). Weather is an example of an external uncertainty, which cannot be influenced. This kind of uncertainty may be modeled stochastically by using a probability distribution, or simply by the parameters' expected values, as in our deterministic model. An example of an internal uncertainty is cargo that never arrives at the port due to miscommunication within the supply chain. The model assumes no such uncertainty, hence a well-functioning supply chain.

The research conducted in Friedberg and Uglane (2012a) revealed some underlying issues in the Logistics' supply chain. Firstly, there is little transparency internally. Together with capacity constraints, this leads to frequent delayed cargo deliveries. Secondly, frequent delays lead to a lack of trust in Logistics. This has lead platform managers to overvalue the prioritization of cargo, an issue that was investigated in Friedberg and Uglane (2012b). These internal uncertainty factors must be addressed before the model will function properly. Logistics have tried to cope with these problems by introducing fixed routes and schedules for PSVs. With fixed schedules, platform managers are able to plan operations more accurately by knowing exactly when the vessel arrives. This approach requires a large fleet of PSVs, which leads to excess capacity on the vessel decks.

If Logistics are able to increase transparency in the supply chain and trust in the system, our model may contribute to better utilization of the PSV fleet by reducing unused space onboard vessels. We have argued that creating routes dynamically based on actual demands is the theoretical best approach to minimizing costs. In addition, the economical study revealed a large potential for reducing direct costs by reducing excess capacity.

Petrobras' 2020 vision aims to double production within 2020. This means that the requirements for effective planning become even more apparent. The pre-salt fields of Santos basin are the main focus of this expansion. As Santos basin is under development, drilling rigs and other non-stationary installations will be the main source for demand. Fixed routes would therefore have to be reset frequently. This kind of environment would fit our model, which is able to create routes dynamically. Also, as distances from the depot increase, refueling offshore might become a necessity for increasing the range of PSVs. The model solved test instances of up to 7 departures, 14 facilities and 6 refueling buoys within a reasonable time frame, which may be considered large with regard to early stages of development in Santos. We therefore believe, with the underlying processes in place, that the model is a promising benchmark on how Petrobras may utilize optimization in the future.

Chapter 9

Conclusion

This thesis has presented the design and implementation of an optimization model for a platform supply vessel routing and scheduling problem. We introduced an operational planning tool customized to fit the current situation of Petrobras Logistics, which supply offshore facilities in Campos basin from the port in Macaé.

We argued that the problem resembles onshore vehicle routing problems with regard to planning horizons, and designed an arc-flow model that was formulated as a vehicle routing problem with pickup and delivery, soft time windows and intermediate refueling tankers. In addition to establishing the routing of PSVs and thereby the sequence in which orders are serviced offshore, the model decides optimal placement of refueling tankers at possible locations in Campos basin. It is our belief after a review of related literature that this is a new aspect within maritime routing problems.

We have emphasized that an operational decision support tool adds value only if it is able to provide good solutions within an acceptable time frame. In an attempt to reduce computational time, the arc-flow model was decomposed by Dantzig-Wolfe principles, which resulted in a path-flow model. A path refers to an entire route for a given vessel and departure. An optimal integer solution was ensured by introducing a Branch and Price scheme.

The two models were implemented in the Xpress-MP system and tested on seven realistic instances from Campos basin. Analysis of the path-flow model revealed that an appropriate column generation strategy contributes to a faster convergence when the root node of the B&P tree is solved. We found that a combined column generation procedure (BFO) proved most efficient. The benefit of BFO is its ability to quickly find columns in the first iterations of the column generation process, thus guiding the RMP in the right direction, and thereafter find optimal columns for the remaining iterations. Furthermore, the technical analyses revealed that the frequency of the IP heuristic did not influence computational times significantly, since the RMP is easily solved compared with the SPs. Although only shown implicitly in our analysis, the decomposed structure of the path-flow model is also a promising basis for parallel computing to further reduce the CPU time.

A comparison of the arc-flow and path-flow models revealed that the path-flow model is superior as the problem size increases. While the arc-flow model leads to large duality gaps and long solution times, the path-flow model is able to reduce the gaps efficiently in the root node of the B&P tree. However, the tail-off effect in the path-flow model is evident as the search continues in the B&P tree. In the CB7 case, the path-flow model had a duality gap of 2.9% after one hour of runtime. After ten hours, this gap was reduced to 1.3%. From a practical planning perspective such small gaps may not be crucial, but the tail-off effect indicates one of the weaknesses of applying B&P.

In the economical study we looked at a base case from Campos basin and utilized the path-flow model to analyse how alterations in the input parameters influence total costs over the planning period. PSVs in Campos currently refuel at the end of their voyage before returning to the depot. The flexibility of being able to refuel any time during the journey did not significantly influence total costs. Similar results were obtained by reducing the fleet of refueling tankers, and by fixing the positioning of a refueling tanker to a non-optimal buoy. These results indicate that the current buoy positions are reasonable in Campos basin.

Regarding alterations in the PSV fleet size and mix the results showed larger differences in cost. By reducing the number of PSV 3000 in the mix, we learned that a capacity deficit lead to a large increase in indirect costs, as it would require express vessels. Furthermore, reducing the number of PSV 1500 resulted in excess capacity, which caused an increase in variable and fixed sailing costs associated with the use of more expensive vessels. These results show that an appropriate PSV fleet size and mix is an important part of the planning process.

Lastly, we made an analysis based on the influence of altering delay penalty. Minor changes in direct costs were observed even with relatively large changes in penalties. We argue that the main contributor to sailing costs is the distance from Macaé to the oil fields; therefore the routing among facilities is not a major contribution to total sailing distance. Consequently, we conclude that minimizing total delay is possible without a significant impact on direct costs.

Logistics are currently facing internal uncertainties in the supply chain and have implemented fixed routes and schedules for PSVs to increase reliability in order handling. This tactical approach to planning may be necessary to provide trust among stakeholders, an element that is required for the supply chain to function properly. In their current state, Logistics may not be compatible with dynamic route planning. However, in relation to the Petrobras 2020 vision and future exploitation and exploration in the pre-salt areas in Santos basin, dynamic route planning could be a valuable asset for more efficient utilization of the PSV fleet. If the company is able to establish trust among stakeholders and transparency along the supply chain, it is our belief that an operational model could be the next step towards further improvements.

Chapter 10

Future Research

This chapter gives a brief overview of possible future research related to the platform supply vessel routing and scheduling problem with refueling tankers. We have discussed and tested different approaches to reduce computational time in the path-flow model. However, not all aspects have been investigated. Solving subproblems by means of parallel computing in Xpress-MP did not result in the desired reduction of solution time. Parallel computing therefore constitutes an interesting topic for further research. For instance, the model could be implemented in software that handles parallel computing more efficiently.

Further enhancements of the Branch and Price algorithm to reduce the tail-off effect could be achieved by utilizing various search and branching strategies. We observed that a correct column generation procedure may reduce total computational time; thus other procedures could be tested. For instance, the possibility of adding multiple columns to the restricted master problem when subproblems are solved may prove efficient. Worth mentioning is the possibility of solving subproblems by the means of dynamic programming, which has proven successful in reviewed literature.

Expanding the model is another interesting topic for further research. Weather is known to be a decisive and uncertain factor for PSVs, which the model does not include. Furthermore, opening hours at offshore installations has been disregarded in the model, which could be a complicating factor if implemented in practice. The aspect of refueling tankers has also been simplified. More elements related to the refueling operations could be implemented, such as the replenishment effect of refueling. The model places a tanker at a single buoy for the entire planning period; another interesting extension is dynamic placement of tankers during the planning period.

Regarding applications of the model, we believe simulating the dynamic route generation process using actual demand data from Campos basin could provide a measure of the actual value of dynamic route planning and scheduling. Total costs could be compared with total costs from the fixed routes that are currently being used by Logistics. Also, evaluating forecast data from Santos basin might be an interesting research area.

Appendix A

Arc-flow model

Sets and indices

$i \in \mathcal{N}$	Set of all nodes, including the depot
$i \in \mathcal{N}^P \subseteq N$	Set of offshore installations
$i \in \mathcal{N}^B \subseteq N$	Set of refueling buoys
$(i, j) \in \mathcal{A}$	Set of arcs
$o \in \mathcal{O}_i$	Set of orders for node i
$o \in \mathcal{O}_i^P$	Set of pickup orders for node i
$o \in \mathcal{O}_i^D$	Set of delivery orders for node i
$v \in \mathcal{V}$	Set of vessel types
$d \in \mathcal{D}$	Set of departures

Parameters

A_o	Area required (pickup and delivery) for order o
B_{ij}	Distance between node i and j
C_v^F	Fixed cost for vessel type v
C_o^P	Penalty per time overdue for order o
C^X	Express delivery penalty
Q_v	Number of vessels for vessel type v available in the planning period
K_v	Capacity (m^2) for vessel type v
L_v^T	Limit on sailing time for vessel type v
L_v^D	Limit on sailing distance for vessel type v
$\overline{T}_o$	Latest start of service for order o without penalty
$\underline{T}_o$	Earliest start of service for order o
T_o^S	Service time at platform upon visit for order o
T_d^D	Departure time for departure d
T^M	Maximum delay allowed
C_{ijv}^V	Sailing cost between node i and j by vessel type v
T_{ijv}^V	Sailing time between node i and j by vessel type v
T^F	Time required for refueling operations
G	Available fleet of refueling tankers

Variables

x_{ijd}	1 if a vessel travels from i to j on departure d, 0 otherwise
y_{od}	1 if order o is delivered on departure d, 0 otherwise
w_o	1 if express delivery is needed for order o, 0 otherwise
t_{id}	Departure time from node i for departure d
s_o	Start time of service for order o
d_o	Arrival delay of order o
l_{id}	Outbound load at node i on departure d
c_d	Variable sailing cost for departure d
v_{vd}	1 if vessel type v is chosen for departure d, 0 otherwise
h_{op}	1 if order o is serviced before order p, 0 otherwise
e_i	1 if a tanker is deployed at node i, 0 otherwise
f_{id}	1 if departure d refuels at node i

Objective function

$$\min z = \sum_{d \in \mathcal{D}} c_d + \sum_{v \in \mathcal{V}} \sum_{d \in \mathcal{D}} C_v^F v_{vd} + \sum_{i \in \mathcal{N}^P} \sum_{o \in \mathcal{O}_i} C_o^P d_o + \sum_{i \in \mathcal{N}^P} \sum_{o \in \mathcal{O}_i} C^X w_o \tag{A.1}$$

Constraints

Variable sailing cost constraints

$$c_d - \sum_{(i,j)\in\mathcal{A}} C^V_{ijv}\, x_{ijd}\, v_{vd} \geq 0 \qquad v \in \mathcal{V}, d \in \mathcal{D} \tag{A.2}$$

Flow network constraints

$$\sum_{j\in\mathcal{N}^P} x_{ijd} \leq 1 \qquad i \in \mathcal{N}, d \in \mathcal{D} \tag{A.3}$$

$$\sum_{i\in\mathcal{N}} x_{ijd} = f_{jd} \qquad j \in \mathcal{N}^B, d \in \mathcal{D} \tag{A.4}$$

$$f_{id} \leq e_i \qquad i \in \mathcal{N}^B, d \in \mathcal{D} \tag{A.5}$$

$$\sum_{i\in\mathcal{N}^B} f_{id} = 1 \qquad d \in \mathcal{D} \tag{A.6}$$

$$\sum_{i\in\mathcal{N}} x_{ijd} - \sum_{i\in\mathcal{N}} x_{jid} = 0 \qquad j \in \mathcal{N}, d \in \mathcal{D} \tag{A.7}$$

$$\sum_{j\in\mathcal{N}} x_{0jd} = 1 \qquad d \in \mathcal{D} \tag{A.8}$$

$$\sum_{i\in\mathcal{N}} x_{i0d} = 1 \qquad d \in \mathcal{D} \tag{A.9}$$

Fleet choice constraints

$$\sum_{v\in\mathcal{V}} v_{vd} = 1 \qquad d \in \mathcal{D} \tag{A.10}$$

$$\sum_{d\in\mathcal{D}} v_{vd} \leq Q_v \qquad v \in \mathcal{V} \tag{A.11}$$

$$\sum_{i\in\mathcal{N}^B} e_i \leq G \tag{A.12}$$

Order balance constraints

$$y_{od} - \sum_{i\in\mathcal{N}} x_{ijd} \leq 0 \qquad j \in \mathcal{N}^P, o \in \mathcal{O}_j, d \in \mathcal{D} \tag{A.13}$$

$$\sum_{o\in\mathcal{O}_j} y_{od} - \sum_{i\in\mathcal{N}} x_{ijd} \geq 0 \qquad j \in \mathcal{N}^P, d \in \mathcal{D} \tag{A.14}$$

$$\sum_{d\in\mathcal{D}} y_{od} + w_o = 1 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{A.15}$$

Order sequencing constraints

$$h_{op} + h_{po} - y_{od} - y_{pd} \geq -1 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i, d \in \mathcal{D} \tag{A.16}$$
$$h_{op} + h_{po} \leq 1 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i \tag{A.17}$$
$$h_{op} - h_{po} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i^D, p \in \mathcal{O}_i^P \tag{A.18}$$

Time constraints

$$\underline{T}_o \leq s_o \leq \overline{T}_o + T^M \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{A.19}$$
$$s_p - (s_o + T_o^S)\, h_{op} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i \tag{A.20}$$
$$t_{0d} = T_d^D \qquad d \in \mathcal{D} \tag{A.21}$$
$$t_{id} - (s_o + T_o^S)\, y_{od} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, d \in \mathcal{D} \tag{A.22}$$
$$s_o - (t_{id} + \sum_{v \in \mathcal{V}} T_{ijv}^V v_{vd})\, x_{ijd}\, y_{od} \geq 0 \qquad i \in \mathcal{N}, j \in \mathcal{N}^P, o \in \mathcal{O}_j, d \in \mathcal{D} \tag{A.23}$$
$$t_{jd} - (T^F + t_{id} + \sum_{v \in \mathcal{V}} T_{ijv}^V v_{vd}) x_{ijd} \geq 0 \qquad i \in \mathcal{N}^P \cap \{0\}, j \in \mathcal{N}^B, d \in \mathcal{D} \tag{A.24}$$

Limit on sailing time

$$t_{id} + \sum_{v \in \mathcal{V}} (T_{i0v}^V - L_v^T)\, v_{vd} \leq T_d^D \qquad i \in \mathcal{N}, d \in \mathcal{D} \tag{A.25}$$

Time delay constraints

$$s_o - d_o \leq \overline{T}_o \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{A.26}$$

Capacity constraints

$$l_{id} - \sum_{v \in \mathcal{V}} K_v\, v_{vd} \leq 0 \qquad i \in \mathcal{N}, d \in \mathcal{D} \tag{A.27}$$
$$\sum_{i \in \mathcal{N}^P} \sum_{o \in \mathcal{O}_i^D} A_o\, y_{od} - l_{0d} = 0 \qquad d \in \mathcal{D} \tag{A.28}$$
$$x_{ijd}\, (l_{id} + \sum_{o \in \mathcal{O}_j^P | i \in \mathcal{N}^P} A_o\, y_{od} - \sum_{o \in \mathcal{O}_j^D | i \in \mathcal{N}^P} A_o\, y_{od} - l_{jd}) = 0 \qquad (i,j) \in \mathcal{A}, d \in \mathcal{D} \tag{A.29}$$

Limit on sailing distance

$$\sum_{(i,j)\in\mathcal{A}} B_{ij}\, x_{ijd} - \sum_{v\in\mathcal{V}} L_v^D v_{vd} \leq 0 \qquad d \in \mathcal{D} \qquad \text{(A.30)}$$

Variable constraints

$$\begin{aligned}
x_{ijd} &\in \{0,1\} && (i,j)\in\mathcal{A}, d\in\mathcal{D} && \text{(A.31)}\\
y_{od} &\in \{0,1\} && i\in\mathcal{N}^P, o\in\mathcal{O}_i, d\in\mathcal{D} && \text{(A.32)}\\
w_o &\in \{0,1\} && i\in\mathcal{N}^P, o\in\mathcal{O}_i && \text{(A.33)}\\
t_{id} &\geq 0 && i\in\mathcal{N}, d\in\mathcal{D} && \text{(A.34)}\\
s_o &\geq 0 && i\in\mathcal{N}^P, o\in\mathcal{O}_i && \text{(A.35)}\\
d_o &\geq 0 && i\in\mathcal{N}^P, o\in\mathcal{O}_i && \text{(A.36)}\\
l_{id} &\geq 0 && i\in\mathcal{N}, d\in\mathcal{D} && \text{(A.37)}\\
c_d &\geq 0 && d\in\mathcal{D} && \text{(A.38)}\\
v_{vd} &\in \{0,1\} && v\in\mathcal{V}, d\in\mathcal{D} && \text{(A.39)}\\
h_{op} &\in \{0,1\} && i\in\mathcal{N}^P, o\in\mathcal{O}_i, p\in\mathcal{O}_i && \text{(A.40)}\\
e_i &\in \{0,1\} && i\in\mathcal{N}^B && \text{(A.41)}\\
f_{id} &\in \{0,1\} && i\in\mathcal{N}^B, d\in\mathcal{D} && \text{(A.42)}
\end{aligned}$$

Linearizations

Variable sailing cost constraints (A.2)

$$\begin{aligned}
c_d - \sum_{(i,j)\in\mathcal{A}} C_{ijv}^V x_{ijd} - M_v^1 v_{vd} &\geq -M_v^1 && v\in\mathcal{V}, d\in\mathcal{D} && \text{(A.43)}\\
M_v^1 &= C_v^S L_v^D && v\in\mathcal{V} && \text{(A.44)}
\end{aligned}$$

Time constraints (A.20)

$$\begin{aligned}
s_p - s_o - (M_{op}^2 + T_o^S)h_{op} &\geq -M_{op}^2 && i\in\mathcal{N}^P o\in\mathcal{O}_i, p\in\mathcal{O}_i && \text{(A.45)}\\
M_{op}^2 &= \max\{s_o - s_p\} && &&\\
&= \overline{T}_o + T^M - \underline{T}_p && i\in\mathcal{N}^P, o\in\mathcal{O}_i, p\in\mathcal{O}_i && \text{(A.46)}
\end{aligned}$$

Time constraints (A.22)

$$t_{id} - s_o - M^3_{op} y_{od} \geq T^S_o - M^3_{op} \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, d \in \mathcal{D} \tag{A.47}$$

$$\begin{aligned} M^3_{op} &= \max\{s_o + T^S_o - t_{id}\} \\ &= \overline{T}_o + T^M + T^S_o - T^D_d - \min_{v \in \mathcal{V}}\{T^V_{0iv}\} \end{aligned} \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, d \in \mathcal{D} \tag{A.48}$$

Time constraints (A.23)

$$\begin{aligned} & s_o - t_{id} - \sum_{v \in \mathcal{V}} T^V_{ijv} v_{vd} \\ & - M^4_{ijod}(x_{ijd} + y_{od}) \geq -2M^4_{ijod} \end{aligned} \qquad i \in \mathcal{N}, j \in \mathcal{N}^P, o \in \mathcal{O}_j, d \in \mathcal{D} \tag{A.49}$$

$$\begin{aligned} & M^4_{ijod} = T^D_d - \underline{T_o} \\ & + \max_{v \in \mathcal{V}}\{L^T_v - T^V_{i0v} + T^V_{ijv}\} \end{aligned} \qquad i \in \mathcal{N}, j \in \mathcal{N}^P, o \in \mathcal{O}_j, d \in \mathcal{D} \tag{A.50}$$

Time constraints (A.24)

$$\begin{aligned} & t_{jd} - t_{id} - \sum_{v \in \mathcal{V}} T^V_{ijv} v_{vd} \\ & - M^5_{ijd} x_{ijd} \geq T^F - M^5_{ijd} \end{aligned} \qquad i \in \mathcal{N}^P \cap \{0\}, j \in \mathcal{N}^B, d \in \mathcal{D} \tag{A.51}$$

$$\begin{aligned} & M^5_{ijd} = T^D_d + T^F \\ & + \max_{v \in \mathcal{V}}\{L^T_v - T^V_{i0v} + T^V_{ijv}\} \end{aligned} \qquad i \in \mathcal{N}^P \cap \{0\}, j \in \mathcal{N}^B, d \in \mathcal{D} \tag{A.52}$$

Capacity constraints (A.29)

$$\begin{aligned} & l_{id} + \sum_{o \in \mathcal{O}^P_j | j \in \mathcal{N}^P} A_o y_{od} - \sum_{o \in \mathcal{O}^D_j j \in \mathcal{N}^P} A_o y_{od} \\ & - l_{jd} + M^6 x_{ijd} \geq M^6 \end{aligned} \qquad (i,j) \in \mathcal{A}, o \in \mathcal{O}_i, d \in \mathcal{D} \tag{A.53}$$

$$\begin{aligned} & l_{id} + \sum_{o \in \mathcal{O}^P_j j \in \mathcal{N}^P} A_o y_{od} - \sum_{o \in \mathcal{O}^D_j j \in \mathcal{N}^P} A_o y_{od} \\ & - l_{jd} + M^6 x_{ijd} \leq M^6 \end{aligned} \qquad (i,j) \in \mathcal{A}, o \in \mathcal{O}_i, d \in \mathcal{D} \tag{A.54}$$

$$M^6 = \max_{v \in \mathcal{V}}\{K_v\} \tag{A.55}$$

Valid inequalities

$$c_d - \sum_{(i,j)\in\mathcal{A}} \min_{v\in\mathcal{V}}\{C^V_{ijv}\}x_{ijd} \geq 0 \qquad d \in \mathcal{D} \tag{A.56}$$

$$\sum_{i\in\mathcal{S}}\sum_{j\in\mathcal{S}} x_{ijd} \leq |\mathcal{S}| - 1 \qquad \mathcal{S} \subseteq \mathcal{N}\backslash\{0\}, |\mathcal{S}| \geq 2, |\mathcal{S}\cap\mathcal{N}^B| \leq 1, d \in \mathcal{D} \tag{A.57}$$

$$\sum_{i\in\overline{\mathcal{S}}}\sum_{j\in\mathcal{S}} x_{ijd} - y_{od} \geq 0 \qquad \mathcal{S} \subseteq \mathcal{N}\backslash\{0\}, |\mathcal{S}| \geq 2, k \in \mathcal{S}, o \in \mathcal{O}_k, d \in \mathcal{D} \tag{A.58}$$

$$t_{id} - \sum_{v\in\mathcal{V}} T^V_{0iv}v_{vd} \geq T^D_d + \min_{o\in\mathcal{O}_i}\{T^S_o\} \qquad i \in \mathcal{N}^P, d \in \mathcal{D} \tag{A.59}$$

$$t_{id} - \sum_{v\in\mathcal{V}} T^V_{0iv}v_{vd} \geq T^D_d + T^F \qquad i \in \mathcal{N}^B, d \in \mathcal{D} \tag{A.60}$$

$$s_o - (T^D_d + \min_{v\in\mathcal{V}}\{T^V_{0iv}\})y_{od} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, d \in \mathcal{D} \tag{A.61}$$

$$t_{id} - (T^D_d + T^F + \min_{v\in\mathcal{V}}\{T^V_{0iv}\})f_{id} \geq 0 \qquad i \in \mathcal{N}^B, d \in \mathcal{D} \tag{A.62}$$

Appendix B

Path-flow model

Master problem

Sets and indices

$r \in \mathcal{R}_{vd}$	Set of routes for departure d and vessel v

Parameters

C^R_{vdr}	Total sailing, fixed and delay costs for vessel v on departure d for route r
C^X	Express delivery penalty
Q_v	Number of vessels for vessel type v available in the planning period
G	Available fleet of refueling tankers
X_{ijvdr}	1 if vessel v travels from i to j on departure d for route r, 0 otherwise
Y_{ovdr}	1 if order o is delivered by vessel v on departure d for route r, 0 otherwise
T_{ivdr}	Departure time from node i for vessel v on departure d for route r
S_{ovdr}	Start time of service for order o by vessel v on departure d for route r
D_{ovdr}	Arrival delay for order o sailing route r on departure d by vessel v
L_{ivdr}	Outbound load at node i for vessel v on departure d for route r
H_{opvdr}	1 if order o is serviced before order p by vessel v on departure d and route r, 0 otherwise
F_{ivdr}	1 if vessel v refuels at node i on departure d and route r, 0 otherwise

Variables

w_o	1 if express delivery is needed for order o, 0 otherwise
e_i	1 if a refueling tanker is deployed at node i
θ_{vdr}	1 if vessel v is chosen on departure d and route r, 0 otherwise

$$\min z = \sum_{v \in \mathcal{V}} \sum_{d \in \mathcal{D}} \sum_{r \in \mathcal{R}_{vd}} C^R_{vdr} \theta_{vdr} + \sum_{i \in \mathcal{N}^P} \sum_{o \in \mathcal{O}_i} C^X w_o \tag{B.1}$$

Constraints

$$\sum_{d \in \mathcal{D}} \sum_{r \in \mathcal{R}_{vd}} \theta_{vdr} \leq Q_v \qquad v \in \mathcal{V} \tag{B.2}$$

$$\sum_{v \in \mathcal{V}} \sum_{d \in \mathcal{D}} \sum_{r \in \mathcal{R}_{vd}} Y_{ovdr} \theta_{vdr} + w_o \geq 1 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{B.3}$$

$$\sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}_{vd}} F_{ivdr} \theta_{vdr} - e_i \leq 0 \qquad i \in \mathcal{N}^B, d \in \mathcal{D} \tag{B.4}$$

$$\sum_{i \in \mathcal{N}^B} e_i \leq G \tag{B.5}$$

$$\sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}_{vd}} \theta_{vdr} = 1 \qquad d \in \mathcal{D} \tag{B.6}$$

$$\theta_{vdr} \geq 0 \qquad v \in \mathcal{V}, d \in \mathcal{D}, r \in \mathcal{R}_{vd} \tag{B.7}$$

$$w_o \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{B.8}$$

$$e_i \geq 0 \qquad i \in \mathcal{N}^B \tag{B.9}$$

Subproblem

Parameters

A_o	Area required (pickup and delivery) for order o
B_{ij}	Distance between node i and j
C_v^F	Fixed cost for vessel type v
C_o^P	Penalty per time overdue for order o
K_v	Capacity (m^2) for vessel type v
L_v^T	Limit on sailing time for vessel type v
L_v^D	Limit on sailing distance for vessel type v
$\overline{T}_o$	Latest start of service for order o without penalty
$\underline{T}_o$	Earliest start of service for order o
T_o^S	Service time at platform upon visit for order o
T_d^D	Departure time for departure d
T^M	Maximum delay allowed
C_{ijv}^V	Sailing cost between node i and j by vessel type v
T_{ijv}^V	Sailing time between node i and j by vessel type v
T^F	Time required for refueling operations
$\overline{\alpha}_v$	Dual variables for constraints (B.2)
$\overline{\beta}_o$	Dual variables for constraints (B.3)
$\overline{\varepsilon}_{id}$	Dual variables for constraints (B.4)
$\overline{\gamma}_d$	Dual variables for constraints (B.6)

Variables

x_{ijvd}	1 if vessel v travels from i to j on departure d, 0 otherwise
y_{ovd}	1 if order o is delivered on vessel v and departure d, 0 otherwise
t_{ivd}	Departure time from node i for vessel v on departure d
s_{ovd}	Start time of service for order o using vessel v and departure d
d_{ovd}	Arrival delay of order o using vessel v and departure d
l_{ivd}	Outbound load at node i on vessel v and departure d
h_{opvd}	1 if order o is serviced before order p by vessel v on departure d, 0 otherwise
f_{ivd}	1 if vessel v refuels at node i on departure d, 0 otherwise

Objective function

$$\begin{aligned}\min p_{vd} = & \sum_{(i,j)\in\mathcal{A}} C_{ijv}^V x_{ijvd} + C_v^F + \sum_{i\in\mathcal{N}^P}\sum_{o\in\mathcal{O}_i} C_o^P d_{ovd} \\ & -\overline{\alpha}_v - \sum_{i\in\mathcal{N}^P}\sum_{o\in\mathcal{O}_i} \overline{\beta}_o y_{ovd} - \sum_{i\in\mathcal{N}^B} \overline{\varepsilon}_{id} f_{ivd} - \overline{\gamma}_d\end{aligned} \tag{B.10}$$

Flow network constraints

$$\sum_{j\in\mathcal{N}^P} x_{ijvd} \leq 1 \qquad i \in \mathcal{N} \tag{B.11}$$

$$\sum_{i\in\mathcal{N}} x_{ijvd} = f_{jvd} \qquad j \in \mathcal{N}^B \tag{B.12}$$

$$\sum_{i\in\mathcal{N}^B} f_{ivd} = 1 \tag{B.13}$$

$$\sum_{i\in\mathcal{N}} x_{ijvd} - \sum_{i\in\mathcal{N}} x_{jivd} = 0 \qquad j \in \mathcal{N} \tag{B.14}$$

$$\sum_{j\in\mathcal{N}} x_{0jvd} = 1 \tag{B.15}$$

$$\sum_{i\in\mathcal{N}} x_{i0vd} = 1 \tag{B.16}$$

Order balance constraints

$$y_{ovd} - \sum_{i\in\mathcal{N}} x_{ijvd} \leq 0 \qquad j \in \mathcal{N}^P, o \in \mathcal{O}_j \tag{B.17}$$

$$\sum_{o\in\mathcal{O}_j} y_{ovd} - \sum_{i\in\mathcal{N}} x_{ijvd} \geq 0 \qquad j \in \mathcal{N}^P \tag{B.18}$$

Order sequencing constraints

$$h_{opvd} + h_{povd} - y_{ovd} - y_{pvd} \geq -1 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i \tag{B.19}$$

$$h_{opvd} + h_{povd} \leq 1 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i \tag{B.20}$$

$$h_{opvd} - h_{povd} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i^D, p \in \mathcal{O}_i^P \tag{B.21}$$

Time constraints

$$\underline{T}_o \leq s_{ovd} \leq \overline{T}_o + T^M \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{B.22}$$

$$s_{pvd} - (s_{ovd} + T_o^S)\, h_{opvd} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i \tag{B.23}$$

$$t_{0vd} = T_d^D \tag{B.24}$$

$$t_{ivd} - (s_{ovd} + T_o^S)\, y_{ovd} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{B.25}$$

$$s_{ovd} - (t_{ivd} + T_{ijv}^V)\, x_{ijvd}\, y_{ovd} \geq 0 \qquad i \in \mathcal{N}, j \in \mathcal{N}^P, o \in \mathcal{O}_j \tag{B.26}$$

$$t_{jvd} - (T^F + t_{ivd} + T_{ivd)}^V)x_{ijvd} \geq 0 \qquad i \in \mathcal{N}^P \cap \{0\}, j \in \mathcal{N}^B \tag{B.27}$$

Limit sailing time

$$t_{ivd} \leq T_d^D - T_{i0v}^V + L_v^T \qquad i \in \mathcal{N} \tag{B.28}$$

Time delay constraints

$$s_{ovd} - d_{ovd} \leq \overline{T}_o \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{B.29}$$

Capacity constraints

$$l_{ivd} \leq K_v \qquad i \in \mathcal{N} \tag{B.30}$$

$$\sum_{i \in \mathcal{N}} \sum_{o \in \mathcal{O}_i^D | i \in \mathcal{N}^P} A_o \, y_{ovd} - l_{0vd} = 0 \tag{B.31}$$

$$\begin{aligned} x_{ijvd} \, (l_{ivd} + & \sum_{o \in \mathcal{O}_j^P | j \in \mathcal{N}^P} A_o \, y_{ovd} \\ - & \sum_{o \in \mathcal{O}_j^D | j \in \mathcal{N}^P} A_o \, y_{ovd} - l_{jvd}) = 0 \end{aligned} \qquad (i,j) \in \mathcal{A} \tag{B.32}$$

Limit sailing distance

$$\sum_{(i,j) \in \mathcal{A}} B_{ij} \, x_{ijvd} \leq L_v^D \tag{B.33}$$

Variable constraints

$$x_{ijvd} \in \{0,1\} \qquad (i,j) \in \mathcal{A} \tag{B.34}$$

$$y_{ovd} \in \{0,1\} \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{B.35}$$

$$t_{ivd} \geq 0 \qquad i \in \mathcal{N} \tag{B.36}$$

$$s_{ovd} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{B.37}$$

$$d_{ovd} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \tag{B.38}$$

$$l_{ivd} \geq 0 \qquad i \in \mathcal{N}, v \in \mathcal{V} \tag{B.39}$$

$$h_{opvd} \geq 0 \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i \tag{B.40}$$

$$f_{ivd} \in \{0,1\} \qquad i \in \mathcal{N}^B \tag{B.41}$$

Linearizations

Time constraints (B.23)

$$s_{pvd} - s_{ovd} - (M^2_{opvd} + T^S_o)h_{opvd} \geq -M^2_{opvd} \qquad i \in \mathcal{N}^P o \in \mathcal{O}_i, p \in \mathcal{O}_i \quad \text{(B.42)}$$

$$\begin{aligned} M^2_{opvd} &= \max\{s_{ovd} - s_{pvd}\} \\ &= \overline{T}_o + T^M - \underline{T}_p \end{aligned} \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i, p \in \mathcal{O}_i \quad \text{(B.43)}$$

Time constraints (B.25)

$$t_{ivd} - s_{ovd} - M^3_{opvd}y_{ovd} \geq T^S_o - M^3_{opvd} \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \quad \text{(B.44)}$$

$$\begin{aligned} M^3_{opvd} &= \max\{s_o + T^S_o - t_{ivd}\} \\ &= \overline{T}_o + T^M + T^S_o - T^D_d - T_{0iv} \end{aligned} \qquad i \in \mathcal{N}^P, o \in \mathcal{O}_i \quad \text{(B.45)}$$

Time constraints (B.26)

$$\begin{aligned} &s_{ovd} - t_{ivd} - T^V_{ijv} \\ &- M^4_{ijovd}(x_{ijvd} + y_{ovd}) \geq -2M^4_{ijovd} \end{aligned} \qquad i \in \mathcal{N}, j \in \mathcal{N}^P, o \in \mathcal{O}_j \quad \text{(B.46)}$$

$$\begin{aligned} &M^4_{ijovd} = T^D_d - \underline{T_o} \\ &+ L^T_v - T^V_{i0v} + T^V_{ijv} \end{aligned} \qquad i \in \mathcal{N}, j \in \mathcal{N}^P, o \in \mathcal{O}_j \quad \text{(B.47)}$$

Time constraints (B.27)

$$\begin{aligned} &t_{jvd} - t_{ivd} - T^V_{ijv} \\ &- M^5_{ijvd}x_{ijd} \geq T^F - M^5_{ijvd} \end{aligned} \qquad i \in \mathcal{N}^P \cap \{0\}, j \in \mathcal{N}^B \quad \text{(B.48)}$$

$$\begin{aligned} &M^5_{ijvd} = T^D_d + T^F \\ &+ L^T_v - T^V_{i0v} + T^V_{ijv} \end{aligned} \qquad i \in \mathcal{N}^P \cap \{0\}, j \in \mathcal{N}^B \quad \text{(B.49)}$$

Capacity constraints (B.32)

$$
\begin{aligned}
& l_{ivd} + \sum_{o \in \mathcal{O}_j^P | j \in \mathcal{N}^P} A_o y_{ovd} - \sum_{o \in \mathcal{O}_j^D j \in \mathcal{N}^P} A_o y_{ovd} \\
& \quad - l_{jvd} + M^6_{ijovd} x_{ijd} \geq M^6_{ijovd} && (i,j) \in \mathcal{A}, o \in \mathcal{O}_i \qquad \text{(B.50)} \\
& l_{ivd} + \sum_{o \in \mathcal{O}_j^P j \in \mathcal{N}^P} A_o y_{ovd} - \sum_{o \in \mathcal{O}_j^D j \in \mathcal{N}^P} A_o y_{ovd} \\
& \quad - l_{jvd} + M^6_{ijovd} x_{ijd} \leq M^6_{ijovd} && (i,j) \in \mathcal{A}, o \in \mathcal{O}_i \qquad \text{(B.51)} \\
& M^6_{ijovd} = K_v && (i,j) \in \mathcal{A}, o \in \mathcal{O}_i \qquad \text{(B.52)}
\end{aligned}
$$

Valid inequalities

$$
\begin{aligned}
& \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S}} x_{ijd} \leq |\mathcal{S}| - 1 && \mathcal{S} \subseteq \mathcal{N} \backslash \{0\}, |\mathcal{S}| \geq 2, |\mathcal{S} \cap \mathcal{N}^B| \leq 1 \qquad \text{(B.53)} \\
& \sum_{i \in \overline{\mathcal{S}}} \sum_{j \in \mathcal{S}} x_{ijd} - y_{od} \geq 0 && \mathcal{S} \subseteq \mathcal{N} \backslash \{0\}, |\mathcal{S}| \geq 2, k \in \mathcal{S}, o \in \mathcal{O}_k \qquad \text{(B.54)} \\
& t_{id} \geq T_d^D + \min_{o \in \mathcal{O}_i} \{T_o^S\} + T_{0iv}^V && i \in \mathcal{N}^P \qquad \text{(B.55)} \\
& t_{id} \geq T_d^D + T^F + T_{0iv}^V && i \in \mathcal{N}^B \qquad \text{(B.56)} \\
& s_o - (T_d^D + T_{0iv}^V) y_{od} \geq 0 && i \in \mathcal{N}^P, o \in \mathcal{O}_i \qquad \text{(B.57)} \\
& t_{id} - (T_d^D + T^F + T_{0iv}^V) f_{id} \geq 0 && i \in \mathcal{N}^B \qquad \text{(B.58)}
\end{aligned}
$$

Appendix C

Digital attachments

The following contents can be found in the attached .zip file:

- The report (Report.pdf)
- LaTeX report folder
- The implemented arc-flow model (ArcFlow.mos)
- The implemented path-flow model (Master.mos and Sub.mos)
- Input files for the Campos basin test cases
- Result files for all analyses conducted in the computational study

Reference List

Aas, B., Gribkovskaia, I., Halskau, Ø., and Shlopak, A. (2007). Routing of supply vessels to petroleum installations. *International Journal of Physical Distribution and Logistics Management*, 37(2):164–179.

Andersson, H., Duesund, J. M., and Fagerholt, K. (2011). Ship routing and scheduling with cargo coupling and synchronization constraints. *Computers and Industrial Engineering*, 61(4):1107–1116.

Aneichyk, T. (2009). Simulation model for strategical fleet sizing and operational planning in offshore supply vessels operations. Master's thesis, Molde University.

Appelgren, L. H. (1969). A column generation algorithm for a ship scheduling problem. *Transportation Science*, 3(1):53–68.

Ashford, R. (2007). Mixed integer programming: A historical perspective with xpress-mp. *Annals of Operations Research*, 149(1):5–17.

Baldacci, R., Battarra, M., and Vigo, D. (2008). Routing a heterogeneous fleet of vehicles. In Bruce Golden, S. R. and Wasil, E., editors, *The Vehicle Routing Problem*, volume 43, chapter 4-27. Springer Science+Business Media, LLC.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., and Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329.

Beasley, J. (1993). *Modern heurstic techniques for combinatorial problems*, chapter 6: Lagrangean relaxation, pages 243–303. Halstad Press.

Beasley, J. E. (1996). *Advances in Linear and Integer Programming*, volume 4 of *Oxford Lecture Series in Mathematics and Its Applications Series*. Oxford University Press, Inc.

Benders, J. F. (1962). Partioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(5):238–252.

Bradley, S., Hax, A., and Magnanti, T. (1977). *Applied Mathematical Programming*. Addison-Wesley.

Business Insider (2012). http://www.businessinsider.com/the-20-most-valuable-energy-companies-in-the-world-2012-1?op=1. Last visited on 10/05/2013.

Christiansen, M. (1996). *Inventory and Time Constrained Ship Routing - A Mathematical Programming Approach*. PhD thesis, Norwegian University of Science and Technology.

Christiansen, M. (1999). Decomposition of a combined inventory and time constrained ship routing problem. *Transportation Science*, 33(1):3–16.

Christiansen, M., Fagerholt, K., Nygreen, B., and Ronen, D. (2005). Maritime transportation. In Barnhart, C. and Laporte, G., editors, *Handbooks in Operations Research and Management Science: Transportation*, volume 14, chapter 4, pages 189–280. Imprint: North-Holland, 1 edition.

Clausen, J. (1999). Branch and bound algorithms-principles and examples. *Department of Computer Science, University of Copenhagen*, pages 1–30.

Colombani, Y. and Heipcke, S. (2006). Multiple models and parallel solving with mosel. *Dash optimization Whitepaper*, 2(9).

Companhia Brasileira de Offshore (2012). http://www.citrosuco.com.br/fischer/. Last visited on 09/05/2013.

Cordeau, F., Laporte, G., and Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52:928–936.

Crevier, B., Cordeau, J.-F., and Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, 176:756–773.

Dantzig, G. B. and Wolfe, P. (1960). Decomposition principle for linear programs. *Operations Research 8*, pages 101–111.

Deif, I. and Bodin, L. D. (1984). Extension of the clarke and wright algorithm for solving the vehicle routing problem with backhauling. *Proceedings of the Babson Conference on Software Uses in Transportation and Logistic Management*, pages 75–96.

Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M. M., and Soumis, F. (2002). *The Vehicle Routing Problem*, chapter 9, pages 225–238. Society for Industrial and Applied Mathematics, 1 edition.

Desrochers, M. and Desrosiers, J. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40:342–354.

Desrosiers, J. and Lubbecke, M. E. (2006). *Column generation*, chapter 1: A primer in column generation, pages 1–32. Springer Science+Business Media, LLC.

Ducrotoy, J.-P., Elliott, M., and de Jonge, V. N. (2000). The north sea. *Marine Pollution Bulletin*, 41(1):5–23.

Elster, K.-H. (1993). *Modern Mathematical Methods of Optimization*, volume 1. Akademie Verlag.

Fagerholt, K. (2001). Ship scheduling with soft time windows: An optimisation based approach. *European Journal of Operational Research*, 131:559–571.

Fagerholt, K., Christiansen, M., Hvattum, L. M., Johnsen, T., and Vabø, T. (2010). A decision support methodology for strategic planning in maritime transportation. *Omega*, 38:465–474.

Farley, A. (1990). A note on bounding a class of linear programming problems, including cutting stock problems. *Operations Research*, 38(5):922–923.

Fisher, M. L. (1994). Optimal solution of vehicle routing problems using minimum k-trees. *Operations Research*, 42(4):626–642.

Fisher, M. L., Jörnsten, K. O., and Madsen, O. B. G. (1997). Vehicle routing with time windows: Two optimization algorithms. *Operations Research*, 45(3):488–492.

Foster, I. (1995). *Designing and building parallel programs*, volume 95. Addison-Wesley.

Fraga, C. d., Borges, F., Bellot, C., Beltrão, R., and Assayag, M. (2003). Campos basin-25 years of production and its contribution to the oil industry. In *Offshore Technology Conference*.

Fried, A. (2011). Supply chain opportunities within petrobras. *Universal Consensus*, pages 36–48.

Friedberg, D. and Uglane, V. (2012a). Integrated planning and logistics. *IO Summer Internship Project*.

Friedberg, D. and Uglane, V. (2012b). Routing and scheduling of platform supply vessels in the brazilian oil industry.

Gélinas, S., Desrochers, M., Desrosiers, J., and Solomon, M. M. (1995). A new branching strategy for time constrained routing problems with application to backhauling. *Annals of Operations Research*, 61(1):91–109.

Goetschalckx, M. and Jacobs-Blecha, C. (1989). The vehicle routing problem with backhauls. *European Journal of Operational Research*, 42:39–51.

Golden, B., Assad, A., Levy, L., and Gheysens, F. (1984). The fleet size and mix vehicle routing problem. *Computers and Operations Research*, 11(1):49–66.

Golden, B., Raghavan, S., and Wasil, E. (2008). *The Vehicle Routing Problem - Latest Advances and New Challenges*, volume 43. Springer Science+Business Media, LLC.

Goodman, R. (2012). World energy insight. In Harris, A., editor, *Energy Leaders' Summit, Istanbul*. World Energy Council.

Gribkovskaia, I. and Laporte, G. (2008). One-to-many-to-one single vehicle pickup and delivery problems. In Golden, B., Raghavan, S., and Wasil, E., editors, *The Vehicle Routing Problem - Latest Advances and New Challenges*, pages 359–377. Springer Science+Business Media, LLC.

Gribkovskaia, I., Laporte, G., and Shlopak, A. (2008). A tabu search heuristic for a routing problem arising in servicing of offshore oil and gas platforms. *Journal of the Operational Research Society*, 59:1449–1459.

Gunnerud, V. (2011). *On Decomposition and Piecewise Lineratization in Petroleum Production Optimization*. PhD thesis, The Norwegian University of Science and Technology, Trondheim, Norway.

Gunnerud, V., Foss, B., Nygreen, B., Vestbø, R., and Walberg, N. C. (2009). Dantzig-wolfe decomposition for real-time optimization - applied to the troll west oil rim. In *ADCHEM Conference*.

Gunnerud, V., Foss, B., and Torgnes, E. (2010). Parallel dantzig–wolfe decomposition for real-time optimization—applied to a complex oil field. *Journal of Process Control*, 20(9):1019–1026.

Halvorsen-Weare, E. E. and Fagerholt, K. (2011). Robust supply vessel planning. *Lecture Notes in Computer Science*, 6701:559–573.

Halvorsen-Weare, E. E., Fagerholt, K., Nonås, L. M., and Asbjørnslett, B. E. (2012). Optimal fleet composition and periodic routing of offshore supply vessels. *European Journal of Operational Research*, 223:508–517.

Ho, J. K., Lee, T. C., and Sundarraj, R. P. (1988). Decomposition of linear programs using parallel computation. *Mathematical Programming*, 42(1-3):391–405.

Igreja, H., Silva, J. R., and Tonidandel, F. (2012). Planning ship operations on petroleum platforms and ports. Contribution to ICKEPS 2012 Challenge Domain.

Kallehauge, B., Larsen, J., and Madsen, O. B. G. (2001). Lagrangean duality applied on vehicle routing with time windows - experimental results.

Kallehauge, B., Larsen, J., Madsen, O. B. G., and Solomon, M. M. (2005). *Vehicle routing problem with time windows*. Springer Science+Business Media, LLC.

Karlof, J. K. (2006). *Integer Programming: Theory and Practice*, chapter 4: Decomposition in Integer Linear Programming. CRC Press.

Kolen, A., Rinnooy Kan, A., and Trienekens, H. (1987). Vehicle routing with time windows. *Operations Research*, 35(2):266–273.

Lemaréchal, C. (2007). The omnipresence of lagrange. *Annals of Operations Research*, 153(1):9–37.

Letocart, L., Touati-Moungla, N., and Nagih, A. (2012). Dantzig-wolfe and lagrangian decompositions in integer linear programming. *Internationl Journal of Mathematics in Operational Research*, 4(3):247–262.

Lundgren, J., Rönnqvist, M., and Värbrand, P. (2010). Optimization. *Studentlitteratur*.

Mitrovic-Minic, S. (1998). Pickup and delivery problem with time windows: A survey. Technical report, Simon Fraser University.

Petrobras Vision for 2020 (2013). http://www.petrobras.com.br/en/about-us/corporate-strategy/. Last visited on 24/05/2013.

Savelsbergh, M. W. P. (1985). Local search in routing problems with time windows. *Annals of Operations Research*, 4:285–305.

Simchi-Levi, D., Kaminsky, P., and Simchi-Levi, E. (2009). *Designing and Managing the Supply Chain: Concepts, Strategies and Case Studies*. McGraw Hill/Irwin, 3 edition.

Stålhane, M., Andersson, H., Christiansen, M., Cordeau, J.-F., and Desaulniers, G. (2012). A branch-price-and-cut method for a ship routing and scheduling problem with split loads. *Computers and Operations Research*, 39(12):3361–3375.

Tarantilis, C. D., Zachariadis, E. E., and Kiranoudis, C. T. (2008). A hybrid guided local search for the vehicle routing problem with intermediate replenishment facilities. *INFORMS Journal on Computing*, 20(1):154–168.

Tebboth, J. R. (2001). A computational study of dantzig-wolfe decomposition. *University of Buckingham*.

The Financialist (2013). Petrobras looks to right the ship. http://www.thefinancialist.com/petrobras-looks-to-right-the-ship/. Last visited on 25/05/2013.

Toth, P. and Vigo, D. (2002). *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, 1 edition.

Vanderbeck, F. and Savelsbergh, M. W. P. (2006). A generic view of dantzig-wolfe decomposition in mixed integer programming. *Operations Research Letters*, 34(3):296–306.

Vanderbeck, F. and Wolsey, L. A. (1996). An exact algorithm for ip column generation. *Operations Research Letters*, 19(4):151–159.

Wallace, S. W. (2003). Decision making under uncertainty: The art of modelling. *Molde University Collage*, 15(3).

World's Energy Resources and Consumption (2010). http://www.consumerenergyreport.com/2010/08/24/energy-resources-consumption/. Last visited on 01/05/2013.

Xpress-MP User's Manual (2006). Xpress-mp user's manual. *Dash Optimization, Northants, UK*.