NTNU – Trondheim
Norwegian University of
Science and Technology

Optimization of a Supply Vessel Scheduling and Fuel Type Allocation Problem for a Hellenic Oil Company

Ingeborg Tveit
Marte Viktoria Øverdal

Industrial Economics and Technology Management
Submission date: June 2013
Supervisor: Kjetil Fagerholt, IØT
Co-supervisor: Marielle Christiansen, IØT

Norwegian University of Science and Technology
Department of Industrial Economics and Technology Management

NTNU
Det skapende universitet

MASTERKONTRAKT

- uttak av masteroppgave

1. Studentens personalia

Etternavn, fornavn	Fødselsdato
Tveit, Ingeborg	**12. aug 1988**
E-post	Telefon
ingebotv@stud.ntnu.no	**48183003**

2. Studieopplysninger

Fakultet	
Fakultet for samfunnsvitenskap og teknologiledelse	
Institutt	
Institutt for industriell økonomi og teknologiledelse	
Studieprogram	Hovedprofil
Industriell økonomi og teknologiledelse	**Anvendt økonomi og optimering**

3. Masteroppgave

Oppstartsdato	Innleveringsfrist
15. jan 2013	**11. jun 2013**
Oppgavens (foreløpige) tittel	
Optimization of a Supply Vessel Scheduling and Fuel Type Allocation Problem for a Hellenic Oil Company	
Oppgavetekst/Problembeskrivelse	
This thesis is devoted to developing an optimization model for the scheduling of fuel supply vessels for a Hellenic oil company. We continue our work from the preceding project assignment, where a basic optimization model was made. The problem is short-sea, considers both contract and spot cargoes and includes time windows. As several, non-mixable types of fuels are demanded, there is also an allocation problem to be considered within each supply vessel.	
Hovedveileder ved institutt	Medveileder(e) ved institutt
Professor Kjetil Fagerholt	**Marielle Christensen**
Merknader	
1 uke ekstra p.g.a påske.	

4. Underskrift

Student: Jeg erklærer herved at jeg har satt meg inn i gjeldende bestemmelser for mastergradsstudiet og at jeg oppfyller kravene for adgang til å påbegynne oppgaven, herunder eventuelle praksiskrav.

Partene er gjort kjent med avtalens vilkår, samt kapitlene i studiehåndboken om generelle regler og aktuell studieplan for masterstudiet.

Trondheim 9/1 -2013
Sted og dato

Ingeborg Tveit
Student

Kjetil Fagerholt
Hovedveileder

Originalen lagres i NTNUs elektroniske arkiv. Kopi av avtalen sendes til instituttet og studenten.

NTNU
Det skapende universitet

MASTERKONTRAKT
- uttak av masteroppgave

1. Studentens personalia

Etternavn, fornavn **Øverdal, Marte Viktoria**	Fødselsdato **03. jan 1988**
E-post **marte.overdal@gmail.com**	Telefon **48218569**

2. Studieopplysninger

Fakultet **Fakultet for Samfunnsvitenskap og teknologiledelse**	
Institutt **Institutt for industriell økonomi og teknologiledelse**	
Studieprogram **Industriell økonomi og teknologiledelse**	Hovedprofil **Anvendt økonomi og optimering**

3. Masteroppgave

Oppstartsdato **15. jan 2013**	Innleveringsfrist **11. jun 2013**
Oppgavens (foreløpige) tittel **Optimization of Fuel Supply Vessel Scheduling for a Hellenic Oil Company**	
Oppgavetekst/Problembeskrivelse This thesis is devoted to developing an optimization model for the scheduling of fuel supply vessels for a Hellenic oil company. We continue our work from the preceding project assignment, where a basic optimization model was made. The problem is short-sea, considers both contract and spot cargoes and includes time windows. As several, non-mixable types of fuels are demanded, there is also an allocation problem to be considered within each supply vessel. The model is aimed to function as a day-to-day planning tool for the oil company, where a rolling planning horizon is utilized.	
Hovedveileder ved institutt **Professor Kjetil Fagerholt**	Medveileder(e) ved institutt **Marielle Christensen**
Merknader **1 uke ekstra p.g.a påske.**	

4. Underskrift

Student: Jeg erklærer herved at jeg har satt meg inn i gjeldende bestemmelser for mastergradsstudiet og at jeg oppfyller kravene for adgang til å påbegynne oppgaven, herunder eventuelle praksiskrav.

Partene er gjort kjent med avtalens vilkår, samt kapitlene i studiehåndboken om generelle regler og aktuell studieplan for masterstudiet.

Trondheim 9/1 -2013
Sted og dato

Marte V. Øverdal
Student

Kjetil Fagerholt
Hovedveileder

Originalen lagres i NTNUs elektroniske arkiv. Kopi av avtalen sendes til instituttet og studenten.

Preface

This is the final thesis of our Master's degree from the Norwegian University of Science and Technology, Department of Industrial Economics and Technology Management. The thesis has been prepared over a period of 20 weeks during the spring of 2013.

The thesis is within the field of operations research and presents a fuel supply vessel scheduling and fuel type allocation problem given by a Hellenic oil company operating in Piraeus Port outside Athens. In order to solve the problem, mathematical models have been developed and implemented, and test results reviewed.

During our work we have had contact with Dr. Nick Rachaniotis, associate Teaching Tutor at the Hellenic Open University and Assistant Professor in Business Administration at Democritus University of Thrace, Department of International Economic Relations and Development. Through e-mail correspondence and during our visit to Athens in January 2013, he has supplied us with useful information about the problem. We are very grateful for all his help, and also for the hospitality we experienced during our visit to Athens. We are also very thankful to our two academic supervisors; Professor Kjetil Fagerholt and Professor Marielle Christiansen. They have provided us with invaluable guidance through all phases in developing this thesis. Thank you very much for all your support and involvement. We would also like to thank Professor Agostinho Agra at the University of Aveiro for taking time to evaluate our model and suggesting possible improvements.

Trondheim, Norway, June 2013

Ingeborg Tveit

Marte Viktoria Øverdal

Abstract

This master thesis is concerned with solving a combined fuel supply vessel scheduling and fuel type allocation problem. The problem is provided by a Hellenic oil company. The company has a small, fixed fleet of fuel supply vessels, which it uses to supply customer ships anchored in the broader area of Piraeus Port outside Athens. The supply vessels are loading multiple types of fuels at refineries before carrying the fuel out to the customer ships. The customers are either mandatory contract ships which must be served, or optional spot ships which may be served if the company has available capacity. The company must decide whether to accept a spot ship or not within only minutes after the inquiry.

Based on the contract customers and the accepted spot customers, the company must generate schedules which specify which customers each supply vessel should supply, when this should be done and in what sequence. The planning horizon is three days. The customers specify a time for when they want to be served, something which must be accounted for in the generated schedules. A given customer ship can place orders of various fuel types to be delivered within the same time interval. All orders placed by a customer ship do not have to be operated by the same vessel, meaning *customer splitting* is possible. The fuels demanded must be allocated to compartments within the supply vessels, and fuels of different types cannot be loaded to the same compartment. Conversely, different orders of the same fuel type may be blended within a compartment, and large orders may be split between several compartments.

The main objective of this thesis is to develop a detailed optimization model for the problem in order to really getting to know the problem and to study its complexity. This model is intended to serve as a starting point for additional research. Further, we want to utilize the model or variants of this to support the company in its decision making.

The model is developed as a mixed-integer programming (MIP) problem and implemented by use of commercial optimization software. The test cases are generated based on real life information from the company. Due to the complexity of the problem, there is made considerable effort in reducing the numbers of variables and constraints in the implementation of the mathematical model. Valid inequalities and model simplifications are added to the basic model with the intention of further improving the model's performance.

The best solutions were obtained by a model which included both tested model simplifications; stowage elimination and no customer splitting. To get feasible solutions with respect to the real allocation problem, the vessels' total capacities had to be set as low as 50 % of the actual vessel capacities. The model with this capacity fraction performed well on test cases of smaller sizes, but in larger test cases this low capacity limits the possibility of obtaining solutions where all demand is met.

The problem is very complex and consequently not easy to solve, but we have seen that simplifying complicating model aspects reduces the model's complexity and makes the model better able to support the company in its decision making. Still, with regards to support the decision making, there is a need for further research.

Sammendrag

Denne masteroppgaven tar for seg et optimeringsproblem som kombinerer ruteplanlegging av forsyningsfartøy med allokering av drivstoff til tanker på fartøyene. Problemet er reelt og er gitt av et gresk oljeselskap. Dette selskapet har en liten flåte bestående av et gitt antall forsyningsfartøy. Disse fartøyene benyttes til å levere drivstoff til større skip som har lagt til i havneområdet ved Piraeus utenfor Athen. Forsyningsfartøyene laster ulike drivstofftyper fra raffinerier før drivstoffet fraktes og leveres til kundeskipene. Noen kundeskip har langtidskontrakter og *må* forsynes av oljeselskapet, mens andre er valgfrie og kan forsynes hvis selskapet har ledig kapasitet. Etter at en ny valgfri forespørsel er mottatt, har selskapet kun kort tid til å avgjøre om de har mulighet til å forsyne kundeskipet eller ikke.

Ut fra de kontraktsfestede og aksepterte kundene lager selskapet ruteplaner som spesifiserer hvilke forsyningsfartøy som skal levere drivstoff til hvilke kundeskip. Ruteplanene inneholder også informasjon om *når* de ulike kundeskipene skal forsynes og i hvilken rekkefølge. En planleggingshorisont på tre dager benyttes i ruteplanleggingen. Ved forespørsel angir kundene et tidspunkt de ønsker forsyning, noe som også må tas hensyn til i planleggingen. Et kundeskip kan etterspørre ordre av flere ulike typer drivstoff til samme tidspunkt. De ulike drivstofftypene som blir etterspurt av et gitt kundeskip kan forsynes av ulike forsyningsfartøy. Det er med andre ord tillatt med såkalt *kundesplitting*. De ulike drivstofftypene må allokeres til tanker på forsyningsfartøyene, og ulike typer drivstoff kan ikke lastes på samme tank. På den andre siden kan ulike ordre av samme drivstofftype allokeres til samme tank, og store ordre kan fordeles over flere tanker.

Hovedmålet med denne masteroppgaven er å utvikle en detaljert optimeringsmodell for problemet for å oppnå god kjennskap til det samt å kunne studere dets kompleksitet. Modellen skal kunne fungere som et utgangspunkt for videre utvikling. Videre ønsker vi å benytte modellen eller varianter av den til å støtte oljeselskapet i sine beslutninger.

Modellen er implementert som et heltallsproblem (MIP) ved bruk av kommersiell optimeringsprogramvare, som videre ble brukt til å løse modellen. Testinstansene er laget med utgangspunkt i informasjon og reelle data fra selskapet. På grunn av problemets kompleksitet er det lagt stor vekt på å redusere antall restriksjoner og variable i implementeringen av modellen. Med

intensjoner om ytterligere forbedringer er modellen også testet med ulike gyldige ulikheter og modellforenklinger.

Den modellen som oppnådde de beste løsningene inkluderte begge de to modellforenklingene; *stuasjeeliminering* og *ingen kundesplitting*. Fartøyenes totalkapasitet måtte reduseres ned til 50 % av den opprinnelige kapasiteten for å få mulige løsninger med hensyn på de faktiske allokeringsrestriksjonene. Modellen med denne kapasitetsandelen ga bra løsninger på de mindre testinstansene, men ved større instanser begrenset den reduserte kapasiteten mulighetene for å få løsninger der alle kundeskipene ble forsynt med drivstoff.

Optimeringsproblemet i denne masteroppgaven er veldig kompleks og dermed vanskelig å løse. Gjennom ulike tester har vi sett at forenklinger av modellens kompliserende faktorer reduserer problemets kompleksitet og øker modellens muligheter til å fungere som støtte til oljeselskapets beslutninger. For å kunne brukes som beslutningsstøtte for oljeselskapet, vil det likevel være nødvendig å utvikle modellen videre.

Contents

1 Introduction

Maritime transportation is a major transportation mode of world trade today, and the volume carried by seaborne trade is growing, (UNCTAD, 2012). Maritime transportation is therefore regarded vital in terms of securing international supply, and the invested funds in the industry are large and increasing by maritime transportation growth. Consequently, even small efficiency improvements may result in large total savings, and taking the right planning and management decisions are therefore important.

Greece is a country with long traditions within the maritime sector. Figure 1 illustrates the number of passengers embarked or disembarked in EU countries in 2011 (European Commission Eurostat, 2013), showing Greece as the second largest country in terms of passenger transportation. Piraeus Port outside Athens is a crucial part of the Hellenic maritime infrastructure, being one of the largest passenger ports in Europe and one of the most important container ports in the Mediterranean Sea. Each year, Piraeus Port serves about 20 million passengers (Piraeus Port Authority S.A., 2013). Figure 2 presents top container ports in the EU in 2011, showing Piraeus on the 11th place (European Commission Eurostat, 2013).

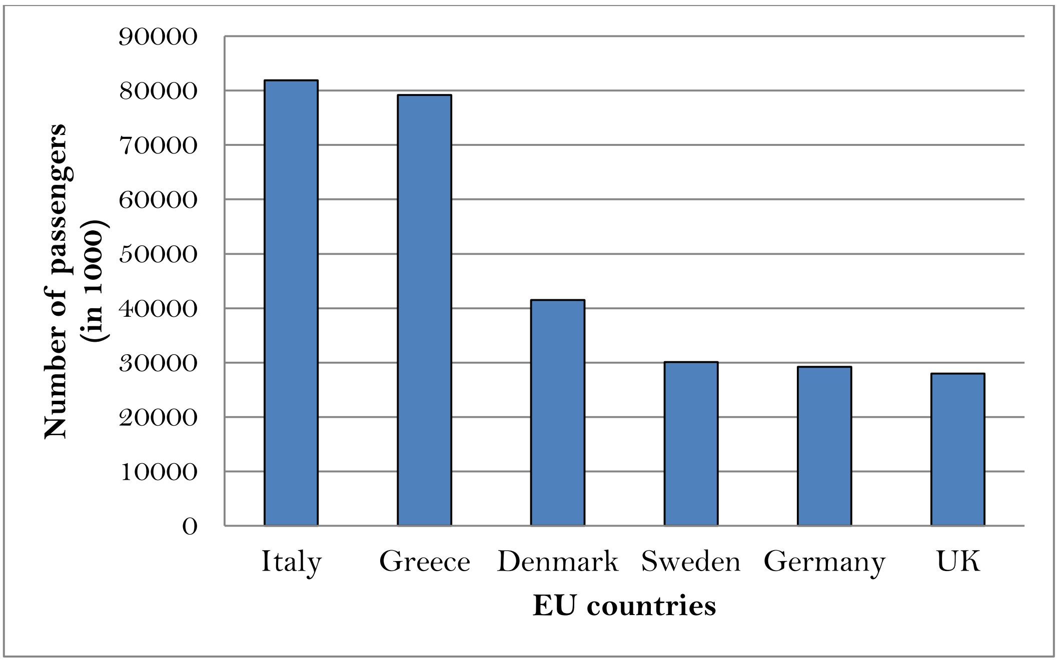


Figure 1: The number of passengers embarked/disembarked in EU countries.

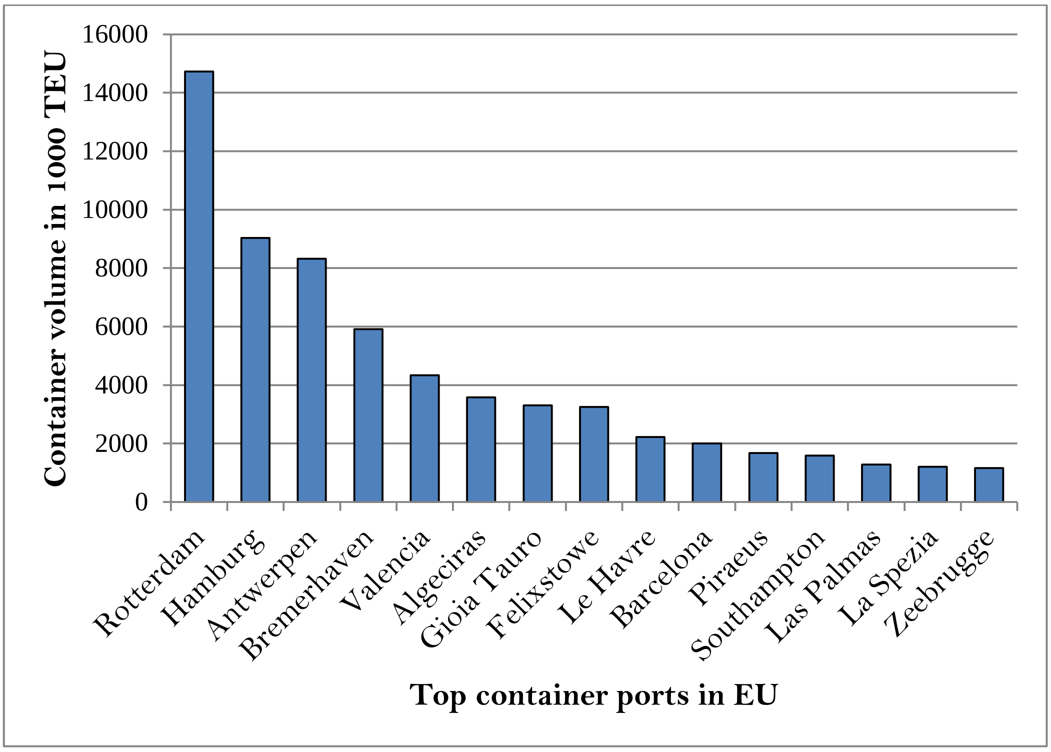


Figure 2: Top container ports in EU, on the basis of volume of containers.

One of the reasons that both passenger and cargo ships enter the port is fuel refilling. The problem studied in this thesis regards this fuel supply business, where incoming ships are supplied with fuels by fuel supply vessels. Figure 3 is a map over the Piraeus area, also showing the area where incoming ships anchor, waiting to be supplied by fuels. The fuel supply business in Piraeus Port has long traditions, and the business is to a large extent characterized by manual efforts in planning decisions. Still, many complicating factors and the large amount of money involved indicate that some technical planning tools could be of good use. Seeing this business in an operations research perspective is therefore very interesting.

The problem in this master thesis was provided by a Hellenic oil company. The company has a small heterogeneous and fixed fleet of fuel oil supply vessels, which the company uses to supply customer ships anchored in the broader area of Piraeus Port. The supply vessels are loaded at the refineries in the inner part of the port before supplying the customer ships. The map in Figure 3 shows the location of the refineries. The refineries offer multiple types of fuel oil, and a customer ship can order quantities of several fuel oil types to be delivered the same day. Some customer ships are mandatory and must be supplied, while other customer ships are optional and can be supplied if the company has the available capacity. The optional orders are called in about three days ahead of delivery time. The company must then decide whether it is possible for them to supply these optional customers, a decision which is not necessarily easy. Based on the mandatory and accepted orders, the vessels' schedules are then generated with regards to maximizing the company's profit. The problem also includes allocating the different types of fuel to separate compartments within the supply vessels, an aspect which adds substantial complexity to the problem.

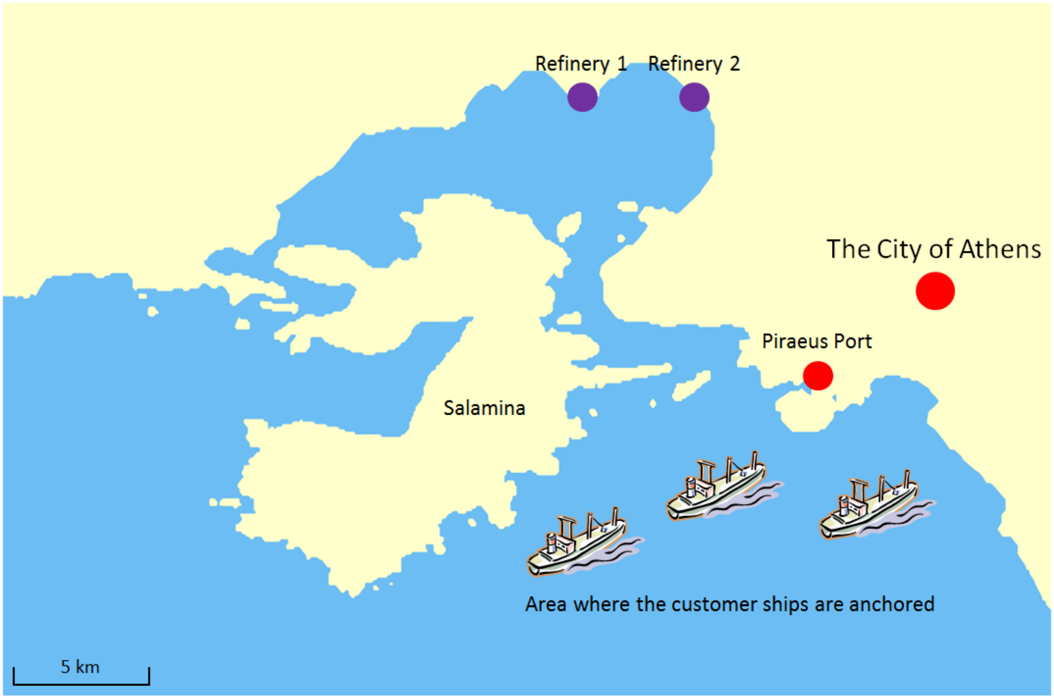


Figure 3: Map of Piraeus port area.

The objective of this thesis is mainly to develop a detailed optimization model for the problem in order to really getting to know the problem and to study its complexity. This model is intended to serve as a starting point for additional research. Further, we want to utilize the model or variants of this to support the company in its decision making.

Since the problem at hand has a very specific nature, we start by a more detailed problem description in Chapter 2. In order to put our problem into a literature context, we will in Chapter 3 discuss how our problem relates to a selection of relevant literature. In Chapter 4 we present the basic mathematical model, which describes all relevant aspects of the real-life planning problem. Chapter 5 deals with the implementation of the mathematical model into commercial programming software. Chapter 6 presents valid inequalities and model simplifications which are applied to the model presented in Chapter 4. The basic mathematical model, the valid inequalities and the model simplifications have been through extensive testing. These tests and the test results are presented in Chapter 7. Finally, Chapter 8 draws some concluding remarks and presents a discussion of what additional research could be done in order to further improve the model.

2 Problem Description

In the problem description we will in Section 2.1 present the fuel supply business in Piraeus Port area. This section describes many definitions and notations used further in this thesis. In Section 2.2 we explain some assumptions made in order to obtain a more general description of the modelled problem. Lastly, we summarize the two sections in Section 2.3. This summary can be seen as a more general description of the problem, which the basic mathematical model will build upon.

2.1 Business Description

About 10 km south of the city centre of Athens is Piraeus Port, one of the largest passenger ports in Europe and of the most important container ports in the Mediterranean Sea. One of the reasons that both cargo and passenger ships enter the port is fuel refilling. There is a small number of competing fuel supply companies that operate in Piraeus Port. They serve both passenger and cargo ships with various fuel types by using different kinds of fuel supply vessels. To distinguish the fuel supply vessels from the cargo and passenger ships in this thesis, the words *vessel* or *supply vessel* are used to denote the fuel supply vessels, while *ship* or *customer ship* denote the cargo and passenger ships that are served by the fuel supply companies.

The fuels that are demanded by the customer ships are loaded onto the supply vessels at two refineries, Elefsina and Aspropyrgos, located in the inner port area of Piraeus Port, as shown in Figure 4. After loading, the supply vessels sail from the inner to the outer port area, where the customer ships are anchored. At each refinery, only two supply vessels can be loaded simultaneously. Because the competing companies load vessels at the same refineries, the refineries are quite busy. The fuel supply companies do not know whether the refineries have available capacity or not, before their vessels arrive there. This makes the loading of the supply vessels very troublesome for the companies to plan. If a company wishes to load one of its vessels, it is common that the vessel must wait several hours outside the refinery because there are already two other vessels from other companies loading. If a refinery has available capacity when a vessel arrives there, the vessel still has to wait for two hours before it can start loading, due to administrative tasks.

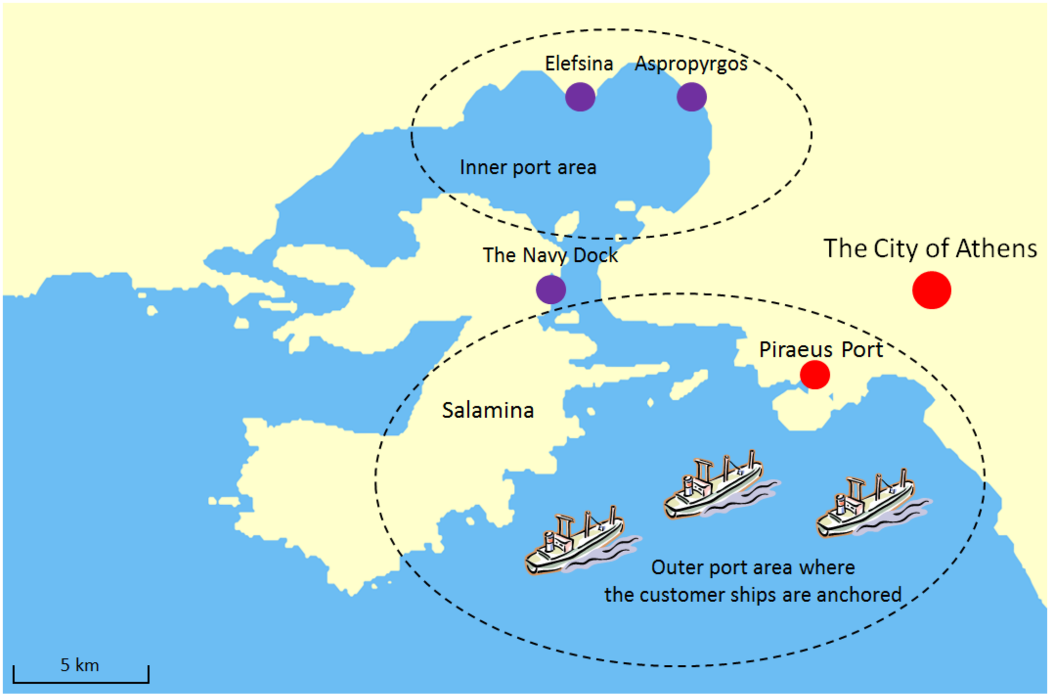


Figure 4: Illustration of the inner and outer port areas. The two refineries are in the inner port area, while the customer ships are anchored in the outer port area.

The Elefsina refinery offers two types of low sulphide fuel oil, while the Aspropyrgos refinery offers two types of high sulphide fuel oil. In addition both refineries offer the same type of gas oil. Summed up, there are five different *fuel types*; four types of *fuel oil* and one type of *gas oil*. It is possible for the supply vessels to be loaded simultaneously with gas oil and a type of fuel oil, as the refineries have separate pumps for gas oil and fuel oils. Overall, 80-85 % of the fuel oils ordered by the customers from the fuel supply companies are high sulphide fuel oils. Ships may place orders for different types of fuel oils and gas oil the same day. At the customer ships, the gas oil is used in generators making electricity and heat, while the fuel oils are used as propellant. Hence, the quantities of gas oil ordered are usually small compared to the quantities of the various fuel oil types. In this thesis, an *order* describes a demanded quantity of one specific fuel type, meaning that one customer ship can place several orders to be delivered the same day.

In this master thesis we will consider one specific fuel supply company operating in the Piraeus Port Area. The company is a Hellenic oil company and will be denoted as *the Oil Company* or *the Company* throughout this thesis. The Oil Company owns a heterogeneous fleet of three supply vessels which operate 24 hours year round. There is a fixed daily cost of using each vessel. In addition, the vessels have different variable sailing costs. The average pumping rate differs between the vessels, and the vessels have up to seven fixed compartments of different sizes. Some compartments may only contain gas oil, while the others can only be used for fuel oils. All three vessels can carry all fuel types, but the various fuel types cannot be blended within a given compartment. Orders of the same fuel type may be allocated to the same compartment, and large orders may be split between compartments. There is no need for cleaning compartments specified for fuel oils between filling them with different types of fuel oils. In this thesis, the word *unload* is used when the supply vessels unload fuel to the customer ships, while *load* is used when the supply vessels load at the refineries. *Operating* is a term that may be used instead of either loading or unloading.

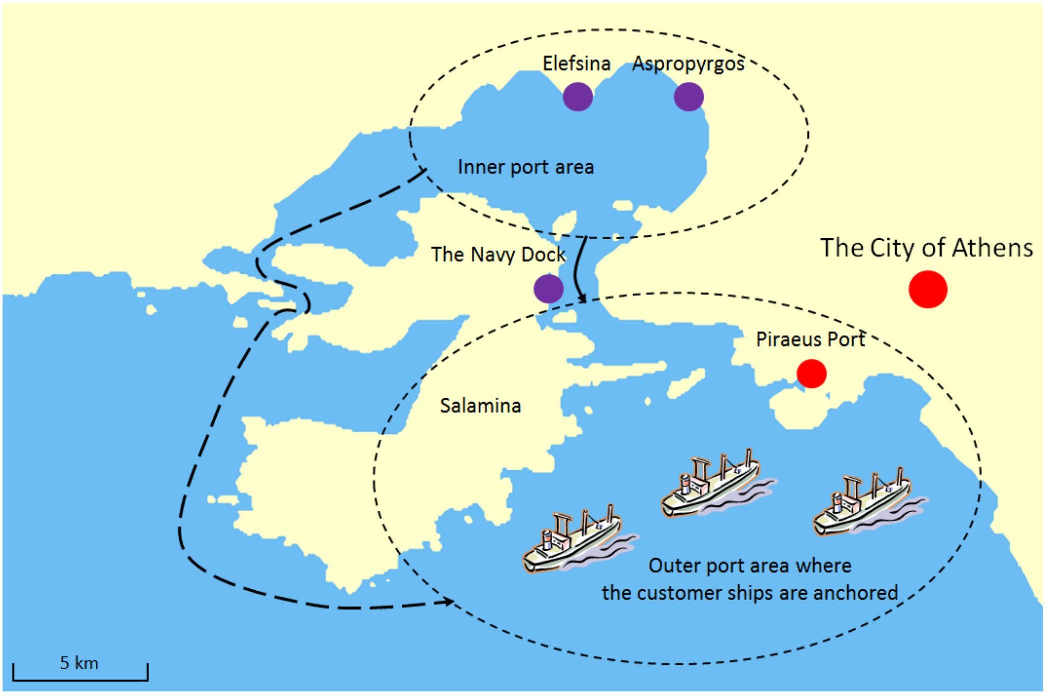


Figure 5: Illustration of the sailing distances between the inner and outer port area. The sailing time between the inner and outer port area increases with three hours at night due to the navy dock closure.

Piraeus Port Area also consists of a navy dock positioned between the inner and outer port areas on the Salamina Island, as shown in Figure 4. Due to security reasons, sailing is not allowed in the area of the navy dock between 21:00 and 6:00, and vessels that want to sail between the inner and outer port area in this period must sail around Salamina Island. This extended sailing distance is shown in Figure 5. There are three considered types of sailing distances; between the two refineries, between a refinery and a customer ship and between the customer ships. Due to the small distances within the Piraeus Port Area, the Oil Company estimates that all distances within the port area have sailing times in the range of 15 to 60 minutes. This relates to all vessels, as the vessels have similar speed rates. The supply vessels' sailing times between a refinery and customer ship, which is sailing between the inner and outer port area, extend by three hours at night due to the *navy dock closure*. Hence, it takes about four hours to sail between the inner and outer port area at night.

The Oil Company has some long-term contracts meaning that some orders are known a long time prior to delivery. The Company is obliged to deliver these contract orders; there is no cancellation possibility, and large penalties occur in case the supply vessels are not able to operate the customers within the agreed time. Nevertheless, most orders are called in about three days ahead of delivery. When a new customer calls, the scheduler must decide to accept or refuse the customer based on the available capacity of the supply vessels. This decision must be taken within about 10 minutes, or else the customer call the competing companies. If the Oil Company decides to operate an optional customer ships, the Company must operate all orders of the given ship. To find out if the Oil Company has the available capacity or not, the fleet scheduler tries to generate a feasible schedule taking into account the newly called in orders. Today, simple "back-of-the-envelope calculations" are used for this. In this thesis, the mandatory contract orders will be denoted *contract orders*, while the optional orders that are called in are referred to as *spot orders*. In the same way, a *contract customer ship* or *contract ship* is a customer ship of contract orders, while a *spot customer ship* or *spot ship* is a customer ship of spot orders. If the Company accepts to serve a spot customer ship, the Company is obliged to deliver its orders in the same way as with contract customer ships.

The scheduler may not necessarily agree on the quantities which the customer originally demands. He might give the customer a new offer that the customer may choose to accept or reject. This offer is typically some fraction of the quantity originally demanded by the customer, and it is something the scheduler may do if he knows that the Company is a bit short on capacity. When the spot orders are accepted by the Oil Company, the spot orders become mandatory contract orders, and the fuel quantity levels are then fixed.

Based on the mandatory and accepted orders, new schedules are generated with regards to maximizing the company's profit. Most customers only specify the date, and not a more specific time, when the orders are to be delivered. Some passenger ships specify that the orders should be delivered in the morning a given date. These time specifications must be taken into consideration in the generation of schedules. The schedules assign customers to the vessels, they give information regarding when and how much of each fuel type the vessels should load which day, and when and in which sequence the customers should be operated. The schedules generated are made for the next three days. Hence, *planning horizons* of three days are usually used by the scheduler. In the generation of a new schedule, the scheduler must take into account tasks which already are allocated to the vessels. Hence, in the new schedule some vessels may not be available for loading until some specified time.

Altogether, the vessels normally operate about six customer ships in total each day. At a given customer ship, only one supply vessel can operate at a time. A customer ship's various orders can be operated by different vessels, although it happens rarely. Usually, the same vessel operate all orders of a given customer ship. The operating time at the customer ship depends on the supply vessel's pumping rate and the fuel quantities ordered by the customer ship, but are normally in the range of one to five hours. Due to the large penalties that occur in case of delays, one of the most important things for the scheduler is to ensure delivery within the agreed time. The scheduler does this by adding slack to the schedules, thus making them more robust. For the unloading part of the schedule, the scheduler plans an operating time that is one third longer than the expected time for unloading the fuels to the customer ships. This time includes the time it takes to attach and detach the pipes of the supply vessels and customer ships. This normally takes only a couple of minutes, but from time to time complications occur. Also, he adds slack to the loading part of the schedule by planning 15 hours for loading each vessel before the vessel start sailing to the customer ships, even though the loading time itself is usually much less than 15 hours. Within these 15 hours he knows that the supply vessel is able to load all the fuel quantities that are to be delivered until next time loading takes place, also if the vessel must visit both refineries. The 15 hours include all necessary loading time in addition to any waiting due to either lack of refinery capacity or administrative matters. The Oil Company's supply vessels never load within the same 15 hours. If the vessels were allowed to load simultaneously, this would increase the traffic at the refineries, possibly increasing the waiting time for each vessel.

2.2 Assumptions

Because time is an important aspect of the real-life problem, we must make some assumptions regarding time in the different problem tasks. These assumptions, and a few additional ones, are presented in this section. All these assumptions are necessary in order to establish a mathematical model of the problem.

The refineries may be considered as a common depot in the inner port area. Within 15 hours it is possible to load all fuel quantities that are to be delivered until next time loading takes place. This means that one vessel may be loaded at both refineries, if required, within these 15 hours. The longest possible time to fully load a vessel is 14.66 hours, which is the time it takes to fully load the largest supply vessel with fuel oil at the refinery with the lowest pumping rate, see Appendix A for calculations. The refinery with the lowest pumping rate is Elefsina, which provides the low-sulphide fuel types. The low-sulphide fuel types amount to only 15-20 % of the total demand; consequently it is unlikely that the largest vessel is fully loaded at this refinery. Also, the vessels may be loaded with fuel oil and gas oil at the same time, meaning that the time it takes to load the small gas oil orders does not have to be taken into consideration. Altogether, this means that we assume 15 hours to be enough time to fully load any supply vessel, including any waiting due to lack of refinery capacity or administrative matters. Based on the current scheduling practice and in order to create a more robust schedule, we assume that none of the Company's vessels can load in the depot simultaneously. The vessels can start loading any time of the day.

If a customer ship places orders to be delivered in the *morning*, we assume that the earliest time of delivery is at 7:00 and all deliveries must be done by 14:00 the specified day. For all other customer ships which only specify a date, we assume that they must be operated between 00:00 and 23:59 this given date. Within these hours operation of all orders at the ships must be finished.

Except from the extended sailing time due to the navy dock closure, all sailing times in the port area vary within the range of 15 to 60 minutes. To make the model robust without making the problem more complex, we assume that all sailing times are equal to 60 minutes. In practice, it is difficult to quickly calculate the exact sailing times as the customer ships may moor or anchor in many different locations in the outer port area. 60 minutes sailing times is therefore a simplifying assumption which also models the problem in an adequate way. Hence; the sailing times are the same no matter if the vessels sail between the inner and outer port area, or between customer ships. As mentioned earlier, the sailing time between the inner and outer port area extends between 21:00 and 06:00 due to the navy dock closure.

Finally we have made some additional assumptions in order to develop the mathematical model:

- A vessel may wait at a customer ship or at the depot before operation, but not after. This is strictly a modelling choice, as it in reality is the same whether a vessel waits before or after operation.
- It is desirable that the orders at a given customer ship are operated continuously, meaning no breaks between the operation of the different orders. If one vessel operates all orders at a customer ship, the orders are operated continuously. If several vessels operate different orders at a customer ship, there is an upper time limit between the start of operation of the first and the last order. The latter is caused by the fact that the vessels have varying pumping rates.
- The vessels always return empty to the depot. Consequently, a vessel only loads the amount of fuel that it shall deliver until next loading, and we assume no order cancellations after the orders are loaded.
- The Company's revenue is correlated to the fuel quantity delivered to spot customers. The decisions to serve contract customers are already made, and income from this part of the business is therefore omitted in the planning objective.
- All kind of uncertainties, for instance uncertainties related to delays, are not considered explicitly. Nevertheless, it is indirectly handled by adding slack to input parameters.

2.3 Summary of the Problem

A small heterogeneous fleet of supply vessels is used to supply customer ships anchored in a port area. The customer ships place orders of different fuel types. The supply vessels load all fuel types in a depot. In the start of a planning horizon, some vessels may not be available for loading until some specified time. After finishing loading at the depot, the supply vessels start sailing to the customer ships. The sailing time between the depot and the customer ships is dependent on the hour of the day. The sailing time between different customer ships are assumed independent of time and which customer ships the vessels sails between. Loading time in the depot is independent of vessel and loading quantity. The depot has a berth capacity which implies that a maximum number of vessels may load at a time. Figure 6 illustrates the customer ships, the supply vessels and the depot. The vessels may wait at a customer ship or at the depot before operation starts.

A vessel's *voyage* starts with loading in the depot, continue with sailing to and operating at the customer ships before returning empty to the depot. For each voyage, a vessel may operate only once in the depot. Within a planning horizon, a vessel can start several voyages. Hence, every time a vessel starts loading in the depot, it also starts a new voyage. In Figure 6, vessel 1 sails two voyages, while vessel 2 only sails one voyage. The vessels must load all quantities to be delivered on the given voyage before leaving the depot.

A customer ship may place orders of different fuel types to be delivered at the same time. Each customer ship states a time interval in which all its orders must be operated. All orders at a customer ship do not need to be operated by the same supply vessel, but if they are, the operation of the orders must happen continuously. If several vessels are operating different orders at the same customer ship, there is an upper time limit between the start of operation of the first and the last order. Also, only one vessel may operate at a customer ship at a time. The supply vessels are obliged to operate contract customers, while spot customers can be operated if the supply vessels have the necessary capacity available. The spot orders' quantities are flexible, but must be within the upper and lower limits specified by the customers. The Company must operate either all or none of the spot customer's orders.

The supply vessels have a different number of compartments where the fuels are held. The compartments are specified for certain fuel types, but each compartment may only contain one fuel type at a time. The same fuel type may be carried in several compartments at the same supply vessel, hence large orders may be split between compartments. Also, if different customer ships order the same fuel type, the orders may be allocated to the same compartment. The vessels' compartments cannot be loaded above their capacity levels.

The objective of the problem is to find solutions that comply with the mentioned constraints, and which maximizes the company's profit. The profit equals the revenue through operation of spot customers subtracted fixed daily costs and variable sailing costs.

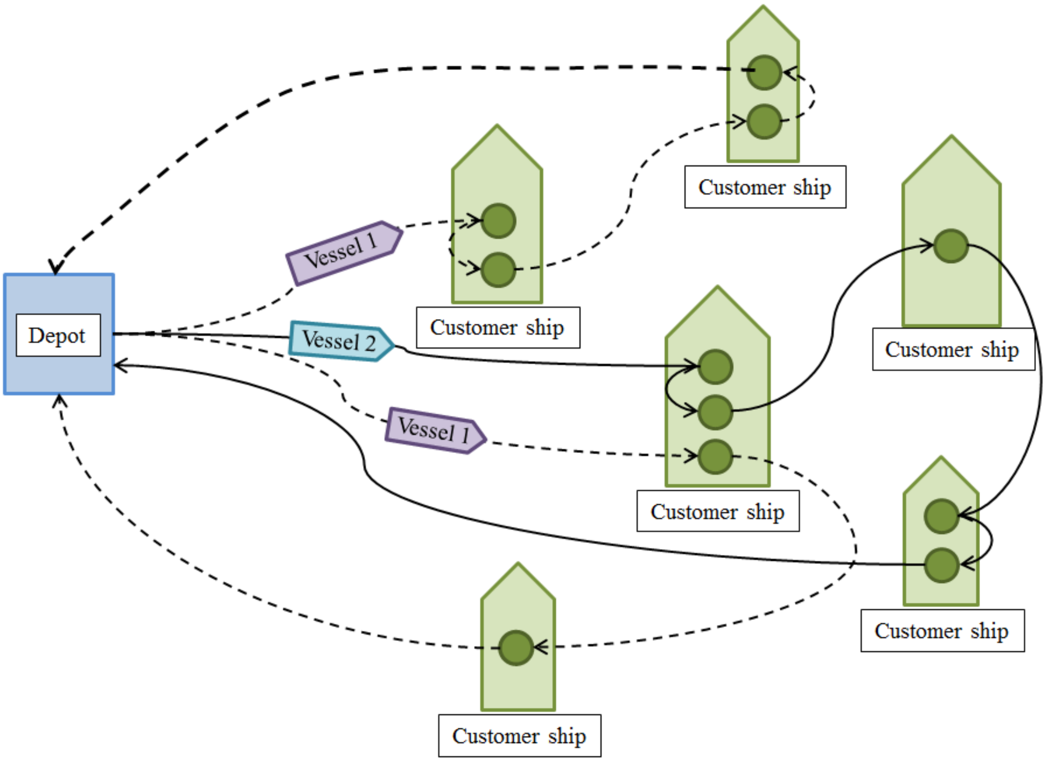


Figure 6: Illustration of the customer ships, vessels and the depot. The costumer ships demand between one and three orders each. The figure illustrates an example with two different vessels and five customer ships in addition to the depot. One of the ships is operated by both vessels, the other ships are operated by only one vessel. Vessel 1 sails two voyages, while Vessel 2 only sails one voyage.

3 Related Literature Review

The focus of this chapter is to put the problem described in Chapter 2 into a literature context. We will do this by looking at important characteristics of our problem and see how existing literature relate to these characteristics.

An extensive overview of maritime transportation is provided by Christiansen et al. (2007). Beside this, surveys of the last decades' research on ship routing and scheduling problems have been published by Christiansen et al. (2013) and Christiansen et al. (2004). These papers have been utilized as a starting point of finding related literature which can be used to set our problem into a literature context. The main part of the studied literature concerns maritime problems, but also other land based operational research problems are studied.

Table 1 shows an overview of how our problem resembles and deviates from the problems in a selection of the investigated literature with respect to various characteristics of our problem. Most of the given characteristics are routing and scheduling characteristics, while the characteristic present on the far right of the table relates to cargo stowage.

Sections 3.1 and 3.2 provide more detailed descriptions of each characteristic presented in Table 1. Together with these descriptions, further descriptions of the listed papers are also included. To give a more complete picture of the various characteristics, some additional literature beyond the papers of Table 1 is also examined.

Table 1: Overview of how investigated literature relates to the characteristics of our problem. The first row represents our problem.

	Maritime VRP	Short Sea Shipping	Uses of Vessels	Time Dependent Sailing Times	Products	Loads	Allocation
Our problem	Yes	Yes	Multiple	Yes	Multiple	Flexible	Yes
Agra et al. (2012a)	No, MIRP	N/A[1)]	Single	No[2)]	Single	Flexible	No
Agra et al. (2012b)	No, MIRP	Yes	Single	No[3)]	Multiple	Flexible	No
Al-Khayyal and Hwang (2007)	No, MIRP	Yes	Multiple	No	Multiple	Flexible	Yes
Brønmo et al. (2007)	No, PDP	N/A[1)]	Single	No	Single	Flexible	No
Christiansen (1999)	No, IPDP	No	Single	No	Single	Flexible	No
Fagerholt and Christiansen (2000)	No, PDP	Yes	Single	No	Multiple	Fixed	Yes
Halvorsen-Weare (2012)	Yes	Yes	Multiple	No	Single	Fixed	No
Hvattum et al. (2009)	No, TAP	N/A[1)]	N/A[1)]	No	Multiple	Flexible	Yes
Kobayashi and Kubo (2010)	No, PDP	No	Single	No	Single	Fixed	Yes
Pang et al. (2011)	No, PDP	Yes	Single	No	Single	Fixed	No

1) No available information.
2) No, but time dependent production and consumption rates.
3) No, but time dependent demand rates and time windows.

3.1 Maritime Routing and Scheduling Characteristics

Our problem can be characterized as a *cargo routing and scheduling problem* following the definition of Al-Khayyal and Hwang (2007). They separate maritime routing and scheduling for bulk products into *cargo routing* and *inventory routing* problems. Cargo routing problems are constrained by specified cargoes to be transported between ports, while inventory routing problems are constrained by inventory levels at ports. Nevertheless, much of the literature concerning *maritime inventory routing problems (MIRP)* share similarities with our problem and are therefore included in this literature review. Since much of the literature within ship routing and scheduling is based upon real industrial applications that mostly vary quite much between the instances, there is a wide variety within the aspects that characterize these problems. As stated by Andersson et al. (2010), a new version is often presented in each paper published.

3.1.1 Maritime VRP

Within ship routing and scheduling most problems are characterized as *pickup and delivery problems (PDP)*, where orders are to be picked up from one port and delivered to another port. Kobayashi and Kubo (2010), Pang et al. (2011) and Fagerholt and Christiansen (2000) describe all such maritime PDPs. A PDP with time windows is called a *PDPTW*. In problems where there is only one common pickup port and many delivery ports, the problem is characterized as a maritime *vehicle routing problem (VRP)*. VRPTW is VRPs with time windows. As the problem name indicates, this type of problem has roots in land based routing problems. Erkut and MacLean (1992) describe a typical land based VRP were food is to be transported by trucks to different stores from a common terminal. By considering the two refineries as a common depot; our problem can be characterized as a maritime VRP where orders are to be picked up from the depot and distributed to the customer ships. Halvorsen-Weare et al. (2012) describe a problem where the offshore supply vessels are used to serve petroleum installations with supplies from an onshore supply depot. Their problem has many similarities to the maritime VRP. Another type of maritime VRP is presented by Dauzère-Pérès et al. (2007). Their problem concerns deliveries of calcium carbonate slurry to European paper manufacturers and besides being a VRP their problem is also an inventory routing problem. Horgen and Frich (2004) describe another maritime VRP which concerns LNG distribution in the Atlantic Basin.

3.1.2 Short Sea Shipping

Many maritime routing and scheduling problems described in the literature concern cargoes that are to be transported long distances between continents. Such a problem is described by Horgen and Frich (2004) and Christiansen (1999), and in these problems the times spent in ports are relatively short compared to the time spent on sailing. Agra et al. (2012b) name their problem a *short sea fuel supply distribution problem*. In their problem, the fuel supply vessels sail the short distances between the islands of Cape Verde. Fagerholt and Christiansen (2000) also describe a short sea problem which is a combined ship scheduling and allocation problem. In their problem, mineral fertilizers are to be transported between production units and discharging ports in Northern Europe. With sailing times in the range from a few hours to almost two days, this represents what is characterized as *short sea shipping*. Within the Piraeus Port Area the various sailing distances are very short, taking maximum a few hours to sail. Hence, our problem can be characterized as a short sea shipping problem. Pang et al. (2011) describe a ship routing problem motivated by an application in which supply vessels shuttle among various terminals in Hong Kong and the Pearl River Delta. Their problem is also a short sea problem, and Pang et al. (2011) point out that loading and unloading time of cargoes at pickup and delivery locations is significant. The main difference between inter-continent and short sea shipping problems is the ratio between the loading/unloading times and the sailing times. The loading/unloading times play a more important role in short sea shipping, and must therefore be modelled more carefully. In inter-continent problems, the operating time may often be neglected.

3.1.3 Multiple Use of Vessels

Azi et al. (2010) give an exact algorithm for a vehicle routing problem with time windows (VRPTW) and multiple use of vehicles. Multiple use of vehicles means that each vehicle may perform several routes during the planning horizon. All routes start and end in the depot. The problem described by Azi et al. (2010), concerns land based routing where the fleet of vehicles is homogenous, meaning all vehicles are equal with respect to for instance capacity and speed. With a homogenous fleet of vehicles, there is no technical difference by using the same vehicle on two routes after each other or using two different vehicles for the same two routes. The fleets in maritime routing problems are often heterogeneous. Hence, the mathematical model of such maritime routing problems must separate between the different vessels. In our problem, we have multiple use of vessels because the vessels may execute several voyages within the planning horizon. In the maritime fuel supply problem described by Halvorsen-Weare et al. (2012), the fleet is heterogeneous due to different deck and bulk capacities of the vessels. The vessels of their problem can also sail multiple voyages within the planning horizon.

3.1.4 Time Dependent Sailing Times

Donati et al. (2006) describe a land based VRP where travel times are time dependent due to variable traffic conditions. They state that accounting for variable travel times is particularly important when planning with time constraints such as time windows. To handle these time dependent parameters, Donati et al. (2006) discretize the time space in a suitable number of subspaces, and include time indices on the variables rather than having time variables. In our problem the sailing time between the inner and outer port area is time dependent due to the navy dock closure at night. Agra et al. (2012a) model a maritime inventory routing problem where the production and consumption rates in ports vary over the planning horizon. Due to these time dependent rates, the models developed by Agra et al. (2012a) have also discrete time formulations. The drawback of using a discrete time approach compared to using a continuous time approach is the increased number of variables. In Agra et al. (2012a) all variables have time-indices, meaning that all variables are generated for every time period of the considered planning horizon. The coarseness of the discretization will affect the number of variables generated. Savelsbergh and Song (2008) model an inventory routing problem with discrete time. They illustrate the trade-off between a reduced problem size with a courser discretization, versus a more detailed description of the real problem with a finer discretization. Agra et al. (2012b) apply a combined discrete and continuous time approach in their model. Discrete time is used by them to handle both time dependent demands and time dependent multiple time windows.

3.1.5 Multiple Products

The inventory routing and fuel supply problem described by Agra et al. (2012a) is a single product problem, meaning that only one type of product is considered. Christiansen (1999) also considers a single product inventory routing problem, where ammonia is to be transported between given ports. Al-Khayyal and Hwang (2007) present a maritime inventory routing problem with multiple products. The quantities delivered to the ports are then specified by both quantity and product type, and the model must ensure that right product type is delivered to the right ports. In their mathematical model, the load variables have own indices representing the product type. In the same way, our problem contains multiple products because the customer ships demand orders of both different quantities and fuel types. Fagerholt and Christiansen (2000) also describe a ship scheduling problem with multiple products. In their problem, different qualities of fertilizers are to be transported from the production ports to the delivery ports.

3.1.6 Flexible Loads

Brønmo et al. (2007) and Brønmo et al. (2010) study a short-term ship routing and scheduling problem with flexible cargo sizes. Because of flexible long-term contracts, the cargo quantity delivered to the specified delivery port must be within a given interval. Having flexible cargo sizes, the cargo quantities are represented by continuous variables instead of fixed parameters in the mathematical model. Korsvik and Fagerholt (2010) provide a heuristic solution method for a general PDPTW with flexible quantities. They state that handling flexible loads is an important short-term routing and scheduling issue. Campbell (2006) describes a VRP with demand range, meaning that the delivered quantity to a customer may deviate from the original demanded size. She states that adding this flexibility to the problem gives potential to generate significant savings in the total distance travelled. The possibility to transport two cargoes on the same vehicle even if the originally demanded quantities together exceed the vehicle's capacity reduces the total travelled distance and will thereby increase the total profit. In our problem, the spot orders are flexible and must be within given intervals, while the contract orders are fixed. In inventory routing problems, such as the problem described be Al-Khayyal and Hwang (2007), flexible loads are very common due to the nature of the problem.

3.2 Cargo Stowage Characteristics

In addition to being a cargo routing and scheduling problem, our problem is also an allocation problem where loads of different fuel types are to be stored in separate compartments. From the listed characteristics of Table 1 the only cargo stowage characteristic is the one named a*llocation*.

Hvattum et al. (2009) describe a *tank allocation problem (TAP)*, a problem of allocating bulk cargoes to tanks in maritime shipping. Their problem is not a routing or scheduling problem, but it only concerns the allocation part for a given route. The vessels considered by Hvattum et al. (2009) have several tanks where the loads can be allocated. They present several constraining aspects that might be important when allocating loads to tanks; constraints against mixing product types in tanks, tank capacity constraints, constraints for minimum load in utilized tanks and constraints concerning product types in neighbouring tanks. In our problem, each vessel contains different compartments of different sizes where multiple fuel types cannot be loaded in the same compartment. In the maritime routing and scheduling problem presented by Kobayashi and Kubo

(2010) each vessel have several fixed compartments, where different cargoes cannot be loaded in the same compartment. Al-Khayyal and Hwang (2007) have dedicated compartments in their problem, meaning that each compartment is dedicated to one product type each. Øvstebø et al. (2011) consider a maritime routing and scheduling problem of RoRo (Roll-on/Roll-off) shipping, where each cargo consists of a set of identical vehicles or other rolling equipment. Only one ship is used to pick up and deliver cargoes, and this ship have a specified capacity in terms of number of decks available for cargoes. The problem described by Øvstebø et al. (2011) has stowage constraints; the ship has stability requirements that must be fulfilled at all times and a given cargo can only be unloaded if no other cargoes block its way out. The ship scheduling problem of Fagerholt and Christiansen (2000) is a combined ship scheduling and allocating problem, where the different qualities of fertilizers cannot be stored together. The transportation ships considered in their problem have flexible sizes of their compartments.

4 Basic Model Description

In this chapter the basic mathematical model of the problem is presented. This model will include all relevant aspects of the real-life problem, and is developed with basis in the summary of the problem given in Section 2.3. Section 4.1 introduces some modelling choices and definitions that are used in the mathematical model. Section 4.2 first gives the indices, sets, parameters and variables, before it presents the objective function and the constraints of the mathematical model. The objective function and the constraints are given together with a more detailed written description.

4.1 Modelling Approach

Even if discrete time representation increases the problem size, as commented in Chapter 3, we have chosen to model time discretely. The main reason for this is the time dependent sailing time between the inner and outer port area, and this parameter must thus have a time index. With discrete time representation, the planning horizon is divided into *time periods* of equal lengths.

In the mathematical model we introduce *nodes* to describe the orders placed by the customer ships. A *node*, *a customer node* and an order is the same, and the terms may be used interchangeably. A customer ship has the same number of nodes as the number of orders it demands. In addition to the nodes representing the orders, we include a depot node and a dummy end node in our model. The depot node represent both the refineries, while the dummy end node represent a fictive node where the vessels end up after operating all scheduled nodes in the planning horizon. The dummy end node is included in the model to better control the flow. In Figure 7, all types of nodes are illustrated.

In Chapter 2 it is stated that a vessel may execute multiple voyages during the planning horizon. In the mathematical model the numbering of voyages is related to each supply vessel. This is illustrated in Figure 7.

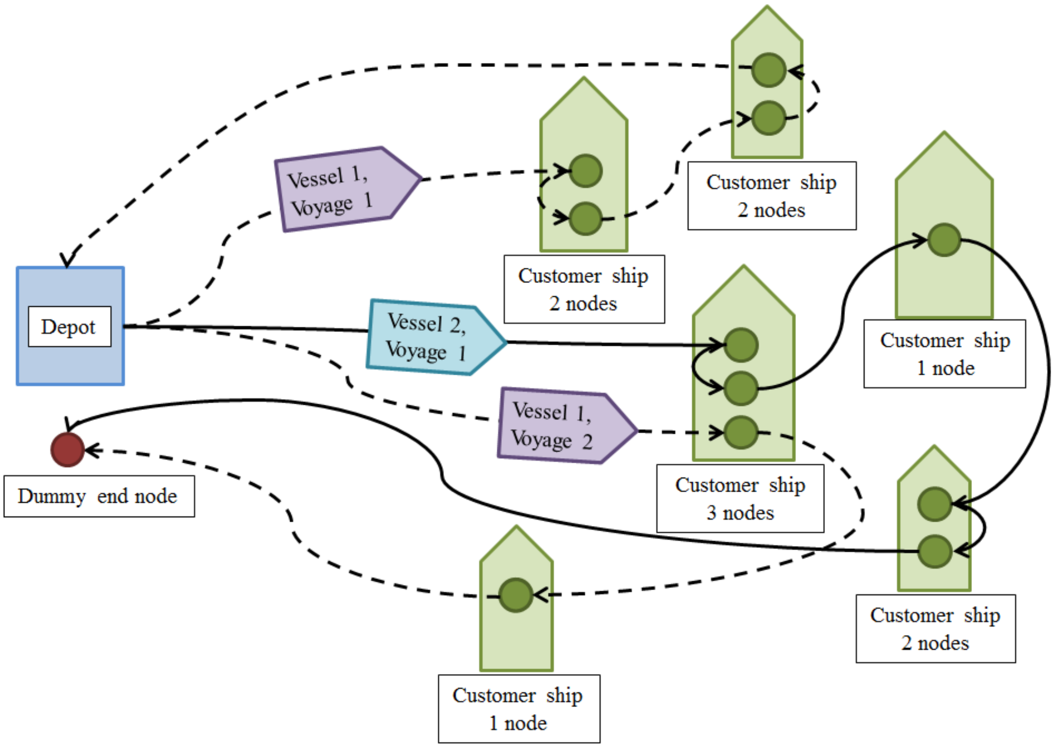


Figure 7: Illustration of the customer ships, vessels, customer nodes, depot node, dummy end node and voyages. Each customer ship has between one and three nodes each. Vessel 1 sails two voyages within the planning horizon before sailing to the dummy end node. Vessel 2 only sails one voyage before ending up at the dummy end node. Note that the sailing to the dummy end node is fictive and not physical sailing.

We also use the concept *time window* in the modelled problem. A customer ship's time window represents the time interval the nodes of the customer ship must be operated in. In this model, the time window of a customer ship is defined by two parameters. One parameter represents the start of the time window. This is the first time period a vessel may start operating one of the customer nodes. The other parameter represents the end of the time window, which is the last possible time period operation at the customer ship may finish. Notice that the time windows here are defined as time periods where operation must be *finished*, while in other relevant literature the time windows are defined as time periods operation may *start*. Figure 8 shows a time line of a customer ship's time window based on the definition used in this thesis. By having time windows related to the customer ships, all nodes at the same customer ship have the same time window.

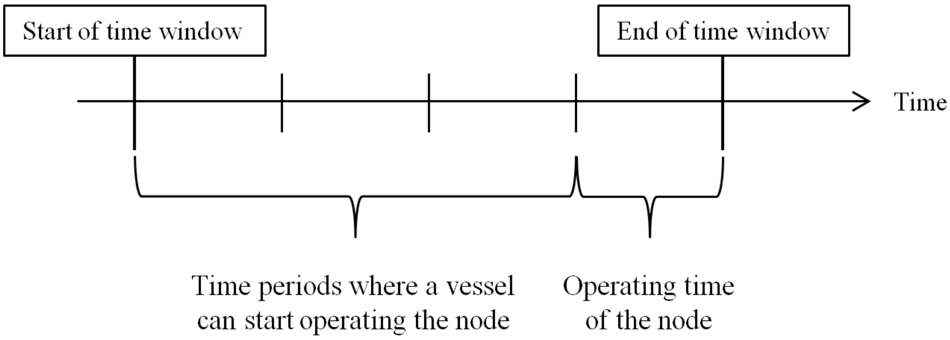


Figure 8: Time line showing a customer ship's time window and the possible time periods where operation can start at a given node of this customer ship.

4.2 The Mathematical Model

4.2.1 The Indices, Sets, Parameters and Variables

Indices

v	supply vessel
i, j	node
0	the depot node
d	the dummy end node
u	customer ship
f	fuel type
c	compartment
m	voyage
t	time period

Sets

$\mathcal{V}$	supply vessels

$\mathcal{N}$	all customer nodes
$\mathcal{A}$	all nodes, $\mathcal{N} \cup \{0\} \cup \{d\}$
$\mathcal{U}$	customer ships
$\mathcal{U}^C \subseteq \mathcal{U}$	contract customer ships
$\mathcal{U}^O \subseteq \mathcal{U}$	optional spot customer ships
$\mathcal{N}_u \subseteq \mathcal{N}$	all nodes that belong to customer ship u
$\mathcal{F}$	fuel types
$\mathcal{F}_c \subseteq \mathcal{F}$	fuel types allowed on compartment c
$\mathcal{C}_v$	compartments on supply vessel v
$\mathcal{M}_v$	set of voyages for vessel v
$\mathcal{T}$	time periods
$\mathcal{T}^{DAY} \subseteq \mathcal{T}$	time periods that represents a day's first time period. For example; when the planning horizon starts with time period 0 and one time period represents one hour, time periods 0, 24, 48 etc. are time periods of this set.
$\mathcal{S}^x$	possible combinations of (v, i, j, m, t) for variable x_{vijmt}
$\mathcal{S}^y$	possible combinations of (v, i, m, t) for variable y_{vimt}
$\mathcal{S}^w$	possible combinations of (v, i, m, t) for variable w_{vimt}

Parameters

T^{SA}_{vijt}	sailing time when vessel v sails directly between node i and j when arriving node j in time period t
T^{SD}_{vijt}	sailing time when vessel v sails directly between node i and j when departing node j in time period t
T^{O}_{vi}	vessel v's operating time at node i

TW_u^{Start}	the start of the time window of customer ship u, the first time period operation may begin
TW_u^{End}	the end of the time window of customer ship u, the last time period that operation may finish
T_u^{MAX}	maximum time between operation can start at the first and the last node at a customer ship u. See Figure 10 for illustration.
T_v^M	the minimum time vessel v may use on any voyage
T_v^E	the earliest time vessel v is available for operation
H	number of time periods within 24 hours
B	berth capacity of the depot
D_{if}	demanded quantity of fuel type f for contract node i
D_{if}^{MIN}	minimum accepted quantity of fuel type f for spot node i
D_{if}^{MAX}	maximum accepted quantity of fuel type f for spot node i
Q_{vc}	load capacity of compartment c on vessel v
C_v^F	fixed daily cost of using vessel v
C_v^S	sailing cost per time period with vessel v
R_f	revenue per quantity delivered of fuel type f

Variables

x_{vijmt}	1, if vessel v starts sailing in time period t from node i directly to node j on voyage m 0, otherwise
y_{vimt}	1, if vessel v starts operating node i in time period t on voyage m 0, otherwise
w_{vimt}	1, if vessel v is waiting in time period t at node i on voyage m 0, otherwise

γ_{vum}	1, if vessel v operates all nodes at customer ship u on voyage m 0, otherwise
z_u	1, if spot customer ship u is operated 0, otherwise
k_{vfcm}	1, if fuel type f is allocated to compartment c of vessel v on voyage m 0, otherwise
δ_{vt}	1, if vessel v is utilized the day that start with time period t 0, otherwise
l_{vijfcm}	quantity of fuel type f in compartment c of vessel v when sailing directly from node i to j on voyage m
q_{vifm}	delivered quantity of fuel type f to spot node i by vessel v on voyage m

4.2.2 Comments to the Formulations of the Mathematical Model

The sailing and operating variables, x_{vijmt} and y_{vimt}, equal 1 if a vessel *start* sailing or operating the given time period. The operation or the sailing itself may take more than one time period. The durations of these activities are given by the sailing time parameters, T^{SA}_{vijt} and T^{SD}_{vijt}, and the operating time parameters, T^{O}_{vi}. The waiting variables, w_{vimt} equal 1 for *each* time period a vessel waits at a node. All these types of variables are illustrated in Figure 9, which is an example of a vessel's flow in a time-space network. The figure illustrates among others that the waiting variable must equal 1 in two following time periods if a vessel waits at a node in two following time periods. Figure 9 also illustrate that the durations of operation and sailing vary.

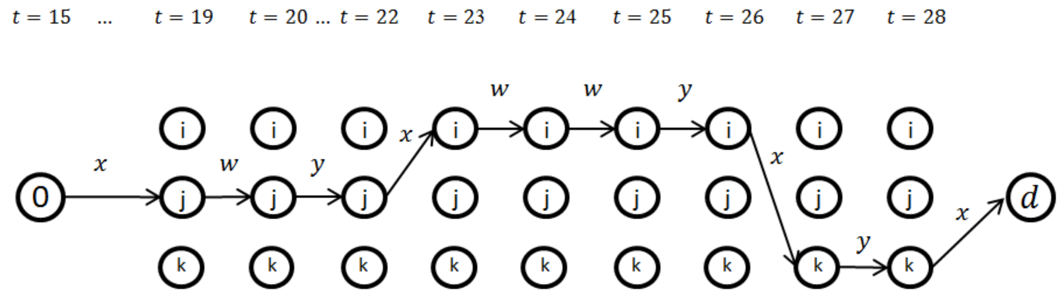


Figure 9: Example of a vessel's flow in a time-space network. The arc labels are y for operating, x for sailing and w for waiting. In this example the vessel starts by sailing from the depot, then it operates at node i, j and k before it sails to the dummy end node. Note that the operating time of node j, T^{O}_{vj}, is 2 time periods, while the operating times of the two other nodes are 1 time period. The sailing from depot to node $j, T^{SA}_{v0j(t=19)}$, is 4 time periods, while the other sailing times in this example are only 1 time period.

In order to improve the model's readability, we have simplified some of the mathematical notations. Some constraints may therefore be defined for whole sets even if this is not quite correct. For instance, constraints (4.5) are defined for all $m \in \mathcal{M}_v$, even though this is only correct for $m \geq 1$. Constraints (4.5) all also defined for all $t \in \mathcal{T}$, but in reality they are only defined for time periods where the vessels are available; $t \geq T^{E}_{v}$.

4.2.3 Objective Function

The objective function (4.1) represents the company's profit. It comprises the revenue from operating spot orders, the daily fixed costs of using the vessels and the variable sailing costs. By including daily fixed costs in this way, the model will strive towards solutions where the vessels are busy some days, and are doing nothing other days. This is assumed to be practical in the real case problem, as long breaks in the utilization of a vessel allow for necessary repairs, time off for the crew and so on.

$$
\begin{aligned}
max\, \Pi = & \sum_{m \in \mathcal{M}_v} \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{N}} \sum_{v \in \mathcal{V}} R_f q_{vifm} \\
& - \sum_{t \in \mathcal{T}} \sum_{v \in \mathcal{V}} C^{F}_{v} \delta_{vt} - \sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}_v} \sum_{j \in \mathcal{A}} \sum_{i \in \mathcal{A}} \sum_{v \in \mathcal{V}} C^{S}_{v} T^{SD}_{vijt} x_{vijmt}
\end{aligned}
\tag{4.1}
$$

4.2.4 Flow Constraints

Constraints (4.2) make sure that every contract node is operated only once, by one vessel on one voyage. The constraints control that the customer nodes are operated within their time windows. Constraints (4.3) hold for the nodes at the spot customer ships. If these nodes are operated, each node can only be operated by one vessel on one voyage, and they must be operated within their time windows. Constraints (4.3) also ensure that all nodes at a given spot customer ship must be operated if the customer ship is operated.

$$\sum_{t=TW_u^{Start}}^{TW_u^{End}-T_{vi}^O} \sum_{m \in \mathcal{M}_v} \sum_{v \in \mathcal{V}} y_{vimt} = 1 \qquad \forall i \in \mathcal{N}_u, u \in \mathcal{U}^C \quad (4.2)$$

$$\sum_{t=TW_u^{Start}}^{TW_u^{End}-T_{vi}^O} \sum_{m \in \mathcal{M}_v} \sum_{v \in \mathcal{V}} y_{vimt} - z_u = 0 \qquad \forall i \in \mathcal{N}_u, u \in \mathcal{U}^O \quad (4.3)$$

As given by the set $\mathcal{S}^y$, the operating variables are only defined in time periods where operation at a given node may begin. The mathematical expression could therefore have been simplified by summing over all time periods. Still, we have chosen the current notation above in order to emphasize the existence of time windows.

Constraints (4.4) ensure that the vessels operate at the depot not more than once on each voyage.

$$\sum_{t \in \mathcal{T}} y_{v0mt} \leq 1 \qquad \forall v \in \mathcal{V}, m \in \mathcal{M}_v \quad (4.4)$$

Constraints (4.5) control that a vessel cannot start a new voyage if it has not started the previous voyage. The constraints also demand that the previous voyage at least takes time T_v^M, which is the minimum time any vessel may use on a voyage.

$$\sum_{\tau=0}^{t-T_v^M} y_{v0(m-1)\tau} - y_{v0mt} \geq 0 \qquad \forall v \in \mathcal{V}, m \in \mathcal{M}_v, t \in \mathcal{T} \quad (4.5)$$

Constraints (4.6) ensure that when a vessel is finished operating a node, it must start sailing to a customer node, the depot node or the dummy end node. Even when the same supply vessel is supplying two different nodes belonging to the same customer ship, it must start sailing after operating the first node. The sailing times between the nodes will be zero in these cases. Since the sailing time between nodes at the same customer ship is zero, sailing variables and operating variables may equal 1 in the same time periods. Sailing to the dummy end node does not represent any physical sailing and it is not possible for a vessel to sail from the depot directly to the dummy end node without operating any customer nodes first. This is ensured by the set $\mathcal{S}^x$, which do not contain combinations of indices for sailing directly from the depot to the dummy end node.

$$y_{vim(t-T_{vi}^{O})} = \sum_{j \in \mathcal{A}} x_{vijmt} \qquad \forall v \in \mathcal{V}, i \in \mathcal{N} \cup \{0\}, m \in \mathcal{M}_v, t \in \mathcal{T} \quad (4.6)$$

Constraints (4.7) make sure that a vessel either starts waiting or operating at a customer node when the vessel arrives the node. Also, if a vessel waits at a node in a time period, it is restricted to either operate or wait at the node in the following time period. Constraints (4.7) hold for customer ship nodes, but it includes sailing from the depot node.

$$\sum_{j \in \mathcal{N} \cup \{0\}} x_{vjim(t-T_{vjit}^{SA})} + w_{vim(t-1)} = y_{vimt} + w_{vimt} \qquad \forall v \in \mathcal{V}, i \in \mathcal{N}, m \in \mathcal{M}_v, t \in \mathcal{T} \quad (4.7)$$

Constraints (4.8) are equivalent to the previous constraints (4.7), but hold for the depot. Constraints (4.8) make sure that when a vessel arrives the depot, it must either start operating the depot on a new voyage or wait at the depot on the current voyage. If a vessel waits at the depot in a time period, it may start operating on a new voyage or keep waiting on the current voyage in the next time period.

$$\sum_{j \in \mathcal{N}} x_{vj0m(t-T_{vj0t}^{SA})} + w_{v0m(t-1)} = y_{v0(m+1)t} + w_{v0mt} \qquad \forall v \in \mathcal{V}, m \in \mathcal{M}_v, t \in \mathcal{T} \quad (4.8)$$

Remember that the sailing time parameter T_{vijt}^{SA} is defined as the sailing time between i and j when *arriving* j in time period t, while the sailing variables

x_{vijmt} equal 1 if vessel v *starts* sailing directly from node i to node j in time period t. This may seem a little strange at first, but letting t in T^{SA}_{vijt} be the arrival time is the most convenient way of expressing sailing time in constraints (4.7) and (4.8).

Constraints (4.9) control that every vessel, if it is used at all, execute the fictive sailing to the dummy end node once during the planning horizon.

$$\sum_{t\in\mathcal{T}} y_{v01t} - \sum_{t\in\mathcal{T}} \sum_{m\in\mathcal{M}_v} \sum_{j\in\mathcal{N}} x_{vjdmt} = 0 \qquad \forall v \in \mathcal{V} \qquad (4.9)$$

Constraints (4.10) ensure that the binary δ_{vt} variables equal 1 if a given vessel is utilized the day which starts with time period t. Waiting is not included, since it is possible to wait at the depot which in practice is not utilizing the vessel. These constraints are included to control the binary δ_{vt} variables used in the objective function (4.1).

$$\sum_{\tau=t}^{t+(H-1)} \sum_{m\in\mathcal{M}_v} \sum_{i\in\mathcal{N}\cup\{0\}} \left(y_{vim\tau} + \sum_{j\in\mathcal{N}\cup\{0\}} x_{vijm\tau}\right) - H\delta_{vt} \le 0$$
$$\forall v \in \mathcal{V}, t \in \mathcal{T}^{DAY} \qquad (4.10)$$

4.2.5 Time Constraints

Constraints (4.11) and (4.12) force the γ_{vum} variables to 1 if all nodes at the same customer ship are operated by the same vessel. Constraints (4.13) further control that the nodes at such customer ships are operated continuously. Note that the constraints assume that the nodes are operated in a specific order, something which reduces symmetry. When a vessel operates all nodes at a customer ship, all nodes must be operated on the same voyage, since continuous operation by the same vessel will never happen on two different voyages.

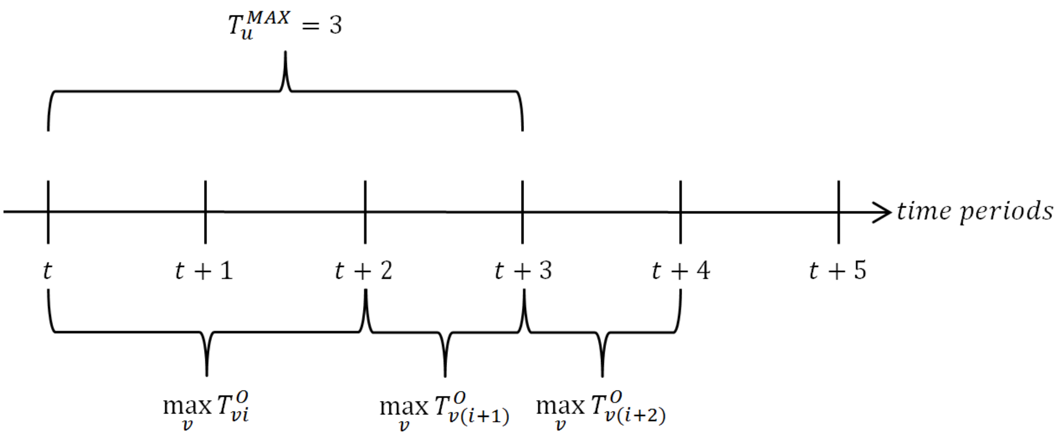
$$\gamma_{vum} - \frac{\sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{N}_u} y_{vimt}}{|\mathcal{N}_u|} \le 0 \qquad \forall v \in \mathcal{V}, u \in \mathcal{U}, m \in \mathcal{M}_v \qquad (4.11)$$

$$\gamma_{vum} - \frac{\sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{N}_u} y_{vimt}}{|\mathcal{N}_u|} \ge \frac{1 - |\mathcal{N}_u|}{|\mathcal{N}_u|} \qquad \forall v \in \mathcal{V}, u \in \mathcal{U}, m \in \mathcal{M}_v \qquad (4.12)$$

$$\left(y_{vimt} - y_{v(i+1)m(t+T^{O}_{vi})}\right) + \gamma_{vum} \leq 1 \qquad \forall v \in \mathcal{V}, i \in \mathcal{N}_u, u \in \mathcal{U}, m \in \mathcal{M}_v, t \in \mathcal{T} \qquad (4.13)$$

If a customer ship is operated by more than one vessel, constraints (4.14) narrow the time span where the nodes at the customer ship can be operated. It is, as described in Chapter 2, desirable that the nodes of a customer ship are operated continuously without any waiting in between. Since the operating times vary with vessel and the fact that the operating sequence of the nodes are not known a priori, these constraints give some possibilities for waiting in between. All operation of nodes at a given customer ship must start within an interval, T^{MAX}_u, calculated from the vessels' operating times at the customer ship. Figure 10 below illustrates how this parameter is calculated.

$$y_{vimt} + y_{v'jn(\tau+T^{MAX}_u)} \leq 1 \qquad \forall v, v' \in \mathcal{V}, i, j \in \mathcal{N}_u | i \neq j, u \in \mathcal{U}, m \in \mathcal{M}_v, n \in \mathcal{M}_{v'}, t, \tau \in \mathcal{T} | \tau > t \qquad (4.14)$$



$D_{if} \geq D_{(i+1)f} \geq D_{(i+2)f} \qquad \forall i, (i+1), (i+2) \in \mathcal{N}_u, u \in \mathcal{U}, f \in \mathcal{F}$

Figure 10: Illustration of the parameter T^{MAX}_u. T^{MAX}_u is illustrated for a given customer ship, u, with three nodes. T^{MAX}_u is the largest possible time difference between start of operation of the first and the last node. This is calculated when the largest order is operated first, the smaller order last, and all orders are operated by the vessel with the lowest pumping rate.

Constraints (4.15) ensure that in any time period, the company cannot have more than B numbers of its vessels loading in the depot. In addition, a customer ship can only be operated by one vessel at the time. Constraints (4.16) take care of the latter.

$$\sum_{\tau=\max\{0,t-T_{v0}^{O}+1\}}^{t} \sum_{m \in \mathcal{M}_v} \sum_{v \in \mathcal{V}} y_{v0m\tau} \leq B \qquad \forall t \in \mathcal{T} \quad (4.15)$$

$$\sum_{\tau=\max\{0,t-T_{vi}^{O}+1\}}^{t} \sum_{m \in \mathcal{M}_v} \sum_{i \in \mathcal{N}_u} \sum_{v \in \mathcal{V}} y_{vim\tau} \leq 1 \qquad \forall u \in \mathcal{U}, t \in \mathcal{T} \quad (4.16)$$

4.2.6 Load Constraints

The difference in load within a supply vessel's compartments before and after operating a customer node equals the demanded fuel quantity of the node. This is ensured by constraints (4.17) for contract nodes and by constraints (4.18) for spot nodes. Constraints (4.19) ensure that the quantity delivered to the spot nodes are within the upper and lower limits. If a spot node is not operated, the quantity delivered will equal zero.

$$\sum_{c \in \mathcal{C}_v} \sum_{j \in \mathcal{N} \cup \{0\}} l_{vjifcm} - \sum_{t \in \mathcal{T}} D_{if} y_{vimt} - \sum_{c \in \mathcal{C}_v} \sum_{j \in \mathcal{A}} l_{vijfcm} = 0 \qquad \forall v \in \mathcal{V}, i \in \mathcal{N}_u, u \in \mathcal{U}^C, f \in \mathcal{F}_c, m \in \mathcal{M}_v \quad (4.17)$$

$$\sum_{c \in \mathcal{C}_v} \sum_{j \in \mathcal{N} \cup \{0\}} l_{vjifcm} - q_{vifm} - \sum_{c \in \mathcal{C}_v} \sum_{j \in \mathcal{A}} l_{vijfcm} = 0 \qquad \forall v \in \mathcal{V}, i \in \mathcal{N}_u, u \in \mathcal{U}^O, f \in \mathcal{F}_c, m \in \mathcal{M}_v \quad (4.18)$$

$$\sum_{t \in \mathcal{T}} D_{if}^{MIN} y_{vimt} \leq q_{vifm} \leq \sum_{t \in \mathcal{T}} D_{if}^{MAX} y_{vimt} \qquad \forall v \in \mathcal{V}, i \in \mathcal{N}_u, u \in \mathcal{U}^O, f \in \mathcal{F}_c, m \in \mathcal{M}_v \quad (4.19)$$

The load variables, l_{vijfcm}, indicate the load on the vessel v when sailing along the arc from node i to node j. Hence, the load variables can be denoted as arc-load flow variables. Agra et al. (2012b) describe the advantages of having arc-load flow variables instead of more common load variables, where the latter do not include a destination node j. They state that using the arc-flow load variables strengthen the model. The drawback is that a larger number of continuous variables are generated.

Constraints (4.20), (4.21) and (4.22) control that the arc-flow load variables, l_{vijfcm}, only are assigned values if a given vessel, v, sails directly from node i to node j. Because of these constraints, summing over all nodes $j \in \mathcal{A}$ in constraints (4.17) and (4.18) do not accumulate the arc-flow load variables. In addition, the constraints (4.20), (4.21) and (4.22) control that the compartments' capacity limits are not exceeded. In constraints (4.21) and (4.22), the quantity supplied to node i may be subtracted from this capacity limit, thus giving a somewhat lower upper limit.

$$\sum_{c\in\mathcal{C}_v}\sum_{f\in\mathcal{F}_c} l_{vijfcm} - \sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}_v} Q_{vc}x_{vijmt} \le 0 \qquad \forall v\in\mathcal{V}, i\in\{0\}, j\in\mathcal{N}, m\in\mathcal{M}_v \tag{4.20}$$

$$\sum_{c\in\mathcal{C}_v}\sum_{f\in\mathcal{F}_c} l_{vijfcm} - \sum_{t\in\mathcal{T}}(\sum_{c\in\mathcal{C}_v} Q_{vc} - \sum_{f\in\mathcal{F}_c} D_{if})x_{vijmt} \le 0 \qquad v\in\mathcal{V}, i\in\mathcal{N}_u, j\in\mathcal{A}, u\in\mathcal{U}^C, m\in\mathcal{M}_v \tag{4.21}$$

$$\sum_{c\in\mathcal{C}_v}\sum_{f\in\mathcal{F}_c} l_{vijfcm} - \sum_{t\in\mathcal{T}}(\sum_{c\in\mathcal{C}_v} Q_{vc} - \sum_{f\in\mathcal{F}_c} D_{if}^{MIN})x_{vijmt} \le 0 \qquad v\in\mathcal{V}, i\in\mathcal{N}_u, j\in\mathcal{A}, u\in\mathcal{U}^O, m\in\mathcal{M}_v \tag{4.22}$$

Constraints (4.23) ensure that only one fuel type is allocated to a compartment on each voyage. The constraints also make sure that a compartment is only loaded with a fuel type that it is allowed to carry. Constraints (4.24) control that the arc-flow load variables only take values for combinations of fuel type and compartment if the fuel type is actually allocated to that compartment. The upper limit of the load variable is the smaller of the compartment capacity limit and the total demanded quantity of the specific fuel type.

$$\sum_{f\in\mathcal{F}_c} k_{vfcm} \le 1 \qquad v\in\mathcal{V}, c\in\mathcal{C}_v, m\in\mathcal{M}_v \tag{4.23}$$

$$l_{vijfcm} - \min\{Q_{vc}, \sum_{k\in\mathcal{N}_u|u\in\mathcal{U}^C} D_{kf} + \sum_{k\in\mathcal{N}_u|u\in\mathcal{U}^O} D_{kf}^{MAX}\}\, k_{vfcm} \le 0 \qquad \forall v\in\mathcal{V}, i,j\in\mathcal{N}\cup\{0\}, f\in\mathcal{F}_c, c\in\mathcal{C}_v, m\in\mathcal{M}_v \tag{4.24}$$

Constraints (4.25) ensure that the vessels do not carry any load when returning to the depot. They also ensure that the vessels contain no load when sailing to the dummy end node.

$$\sum_{c \in \mathcal{C}_v} \sum_{f \in \mathcal{F}_c} \sum_{j \in \mathcal{N}} l_{vjifcm} = 0 \qquad \forall v \in \mathcal{V}, i \in \{0\} \cup \{d\}, m \in \mathcal{M}_v \tag{4.25}$$

4.2.7 Variable Constraints

Constraints (4.26) - (4.34) describe the variable restrictions. The spot quantity variables and the arc-load flow variables are continuous variables, while the other variables are subjects to binary requirements. The continuous variables are restricted by non-negativity constraints.

$$x_{vijmt} \in \{1,0\} \qquad \forall (v,i,j,m,t) \in \mathcal{S}^x \tag{4.26}$$

$$y_{vimt} \in \{1,0\} \qquad \forall (v,i,m,t) \in \mathcal{S}^y \tag{4.27}$$

$$w_{vimt} \in \{1,0\} \qquad \forall (v,i,m,t) \in \mathcal{S}^w \tag{4.28}$$

$$z_u \in \{1,0\} \qquad \forall u \in \mathcal{U}^O \tag{4.29}$$

$$\gamma_{vum} \in \{1,0\} \qquad \forall v \in \mathcal{V}, u \in \mathcal{U}, m \in \mathcal{M}_v \tag{4.30}$$

$$\delta_{vt} \in \{1,0\} \qquad \forall v \in \mathcal{V}, t \in \mathcal{T}^{DAY} \tag{4.31}$$

$$k_{vfcm} \in \{1,0\} \qquad \forall v \in \mathcal{V}, f \in \mathcal{F}_c, c \in \mathcal{C}_v, m \in \mathcal{M}_v \tag{4.32}$$

$$l_{vijfcm} \geq 0 \qquad \forall v \in \mathcal{V}, i \in \mathcal{N} \cup \{0\}, j \in \mathcal{A}, f \in \mathcal{F}_c, c \in \mathcal{C}_v, m \in \mathcal{M}_v \tag{4.33}$$

$$q_{vifm} \geq 0 \qquad \forall v \in \mathcal{V}, i \in \mathcal{N}, f \in \mathcal{F}, m \in \mathcal{M}_v \tag{4.34}$$

5 Implementation

This chapter describes the implementation of the basic mathematical model into commercial software. The model has been implemented in Mosel and solved using the optimization software Xpress v7.3 64-bit. All computational tests have been run on an HP DL 165 G6 computer with two AMD Opteron 24312 4.0 GHz processors, 24 GB of RAM and running on a Linux operating system. Even though the processors used to run these tests have multiple cores, only single thread versions of the programs have been run, to give running times comparable to using a single core computer. Section 5.1 describes the implemented model's structure. Section 5.2 presents how the variables and constraints are created with respect to reducing the problem size. Section 5.2 also includes pseudo codes which illustrate how variables are created in Mosel. The implemented model is attached to the master thesis in own files.

5.1 The Implemented Model's Structure

The basic mathematical model is implemented in a single Mosel file. All constraints and variables of the mathematical model are declared and created in this file. Data input is given to the Mosel file from a text file and an Excel file. Fixed data, such as the number of vessels, their compartment capacities and pumping rates, the number of fuel types and a table of which fuel type that can be loaded in which compartments are given in the text file. The information concerning the customer ships are given in the Excel File. The Excel file contains information regarding the number of customer ships the model have to deal with, their demanded quantities and fuel types, the start of their time windows and whether they are spot or contract ships. Based on the information from the text file and the Excel file, all remaining parameters are calculated in the Mosel file.

In Chapter 7, computational studies of the different model tests will be presented. When testing on cases with a varying number of customer ships, different Excel files are used as input to the Mosel file. In a single test run, there is always one text file and one Excel file used as input files to the Mosel file. If a test requires changes in constraints or changes in subscripts of the variables, a different Mosel file based on the original one is used. If the model changes force changes in the fixed data as well, an altered text file will be used as input file.

5.2 Creating Variables and Constraints

The complexity and size of the problem depend a lot on the numbers of binary decision variables and constraints. In the problem matrix, each variable represents a column, while each constraint represents a row. As stated in Chapter 4, the variables in the mathematical model do not exist for every subscript combination. In the implementation of the basic model it is put much effort in reducing the number of variables by only create variables with possible subscript combinations. To avoid creation of unnecessary empty rows, an effort is also made in the creation of the constraints. Such a comprehensive variable and constraint reduction is done in order to avoid that the computer runs out of memory before any solutions are achieved.

To express, among others, the time periods where the vessels can sail out from the depot, there is a need for a sailing time parameter which is independent of the (v, i, j, t) indices. Hence, we introduce a new sailing time parameter, T^S. This parameter represents the minimum sailing time between the inner and outer port area. Remember that the refineries are placed in the inner, while all customer ships are placed in the outer port area. The minimum sailing time, T^S, occurs in time periods where the navy dock is open. From the assumption of Section 2.2, all sailing times within the port area are equal when the navy dock is open. With this, the sailing time T^S also represents the sailing time between the customer nodes.

To simplify the expressions of time periods where the different nodes can be operated, we introduce time window parameters for the nodes, TW_i^{Start} and TW_i^{End}. These parameters also increase the readability of the given pseudo codes. In addition, we introduce time window parameters for the depot node and the dummy end node. It is not possible to operate the dummy end node, but the time window parameters are still defined to control when it is possible to sail to this node. In the mathematical model, the time window parameters TW_u^{Start} and TW_u^{End} represent the start and end of a customer ship's time window. For a given node i at customer ship u, $i \in \mathcal{N}_u$, the time window parameters will be equal; $TW_u^{Start} = TW_i^{Start}$ and $TW_u^{End} = TW_i^{End}$.

5.2.1 Creating the Sailing, Operating and Waiting Variables

The sets $\mathcal{S}^x$, $\mathcal{S}^y$ and $\mathcal{S}^w$ contain the possible subscript combinations for respectively the sailing, operating and waiting variables. As mentioned, only variables with these possible combinations of subscripts are implemented. For all these variables three different time aspects restrict the possible subscript combinations. In this section, each aspect will be described sequentially before we give a description of how the aspects are combined.

Aspect 1
Operating at a node, either a customer ship node or the depot can only be done within the node's time window. Since the operating variables, y_{vimt}, equal one if vessel v on voyage m *starts* operating node i in time period t, the operating variables are only created for (v, i, m, t) combinations which ensure operation to be finished within the node's time window. Remember that T_{vi}^{O} is vessel v's operating time at node i. For all voyages, $m \in \mathcal{M}_v$, an operating variable y_{vimt} is created if:

$$t \geq TW_i^{Start} \tag{5.1}$$

and

$$t \leq TW_i^{End} - T_{vi}^{O} \tag{5.2}$$

Sailing from a customer node or from the depot can only occur after operation of the respective node, while waiting at a node can only occur before operating. Hence, the creation of the sailing and the waiting variables are also done with respect to the nodes' time windows. When creating the sailing variables, x_{vijmt}, the time windows of both the departure node i and the destination node j must be taken into consideration. Remember that the sailing variables, x_{vijmt}, equal one if vessel v on voyage m *starts* sailing from departure node i directly to destination node j in time period t. It is not possible to leave the departure node after its time window, neither is it allowed to arrive the destination node after its time window. In addition it must be possible to operate the destination node before its time window ends. For all voyages, $m \in \mathcal{M}_v$, a sailing variable x_{vijmt} is created if:

$$t \geq TW_i^{Start} + T_{vi}^{O} \tag{5.3}$$

and

$$t \leq \min(TW_i^{End}, TW_j^{End} - T_{vj}^{O} - T^{S}) \tag{5.4}$$

Since waiting only may happen before operating, a vessel cannot wait at a node after the last possible time period where operation can start. The start of the node's time window does not limit the number of created (v, i, m, t)

combinations because it is possible to wait at a node before its time window starts. For all voyages, $m \in \mathcal{M}_v$, a waiting variable w_{vimt} is created if:

$$t < TW_i^{End} - T_{vi}^O \tag{5.5}$$

Aspect 2

As described in the mathematical model by constraints (4.5), it is not allowed for a vessel to start a new voyage before the previous is finished. The depot must be operated on every voyage. Hence, a customer ship node cannot be operated on a vessels' voyage before the vessel is finished with operating the depot on its current and the previous voyages. In addition the vessel must have had time to sail the necessary number of times between the depot and the customer nodes and operated a minimum number of customer nodes. For instance; for a given vessel at the second voyage, the depot cannot be operated before the vessel has operated the depot on the first voyage, operated at least one customer node on the first voyage and sailed from the depot and back to the depot one time each. Figure 11 illustrates this example. The earliest time operating can start at node i on the second voyage is illustrated on a time line in Figure 12. Based on the reasoning above, an operating variable $y_{vimt} | i \in \mathcal{N}$ is created if:

$$t \geq T_v^E + mT_{v0}^O + (2m-1)T^S + \min_{j \in \mathcal{N} | j \neq i} (m-1)T_{vj}^O \tag{5.6}$$

Note that the earliest time vessel v is available for operation, T_v^E, is also a part of the expression. A vessel cannot start operating in the depot until the time period where it is available. For the depot the expression of the lower bound of the time indices differ some from the case above with customer ship nodes. It is only required that the depot node has been operated on the *previous* voyages before operating, not on the current. In addition, the expression includes one sailing distance. Notice that the expression represents the definition of the minimum duration of a voyage, given as T_v^M in the mathematical model, see Figure 11 for illustration. The operating variables, y_{v0mt}, are created if:

$$t \geq T_v^E + (m-1)T_{v0}^O + 2(m-1)T^S + \min_{(j \in \mathcal{N})} (m-1)T_{vj}^O \tag{5.7}$$

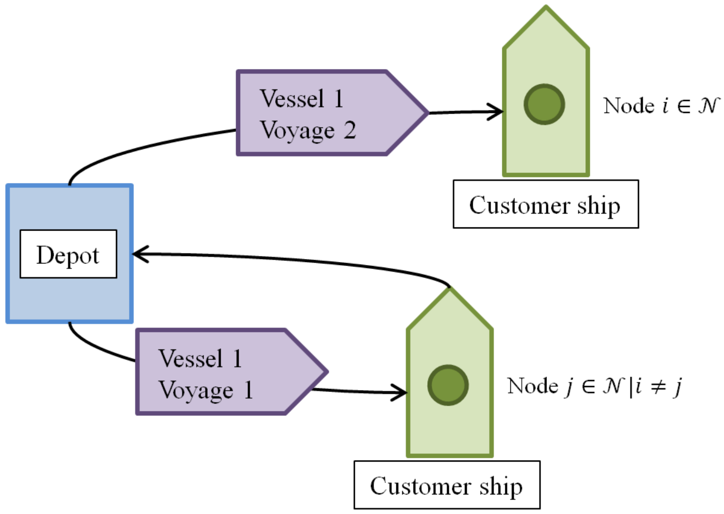
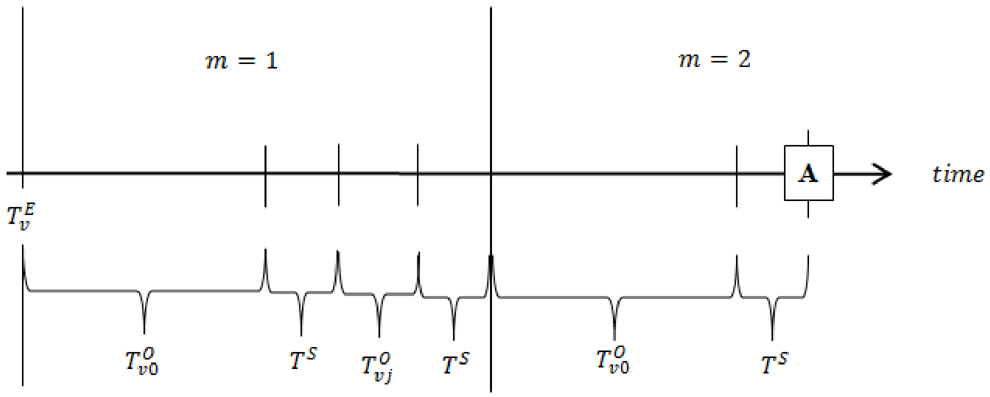


Figure 11: Illustration of Aspect 1, showing the moves and activities vessel 1 must execute before operating node i on voyage 2. Voyage 1 by vessel 1 is a voyage with only one customer node operated. The minimum operating time of any node $j|(j \neq i)$ is $\min_{j \in \mathcal{N}} T^O_{vj}$. With this node j operated as the only one on voyage 1, the duration of this voyage equals T^M_v.



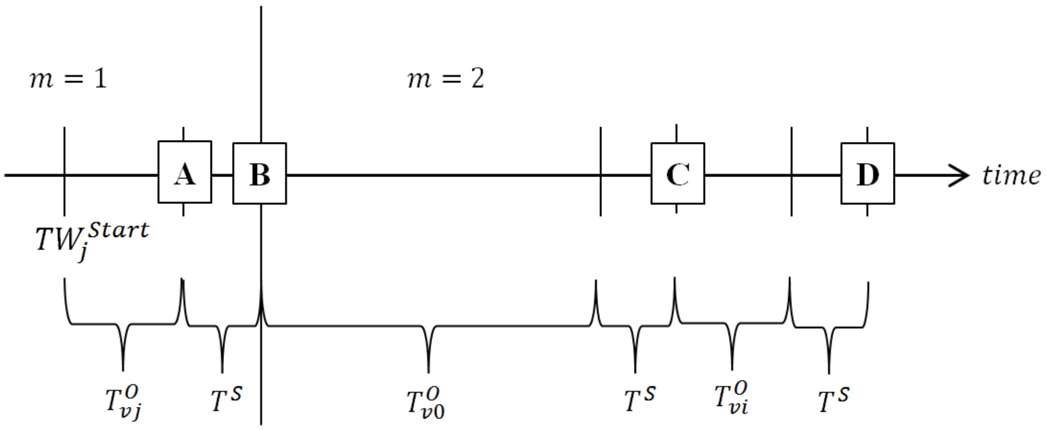
A: Earliest time node i can be operated by vessel v on voyage $m = 2$.

Figure 12: Illustration of Aspect 1, showing a time line which indicates the earliest time node i can be operated by a vessel v on voyage 2. Node j is operated on the first voyage by the same vessel.

Aspect 3

The last of the aspects which limit the numbers of possible (v, i, m, t) combinations of the operating, sailing and waiting variables is very problem specific. As will be described in Chapter 7, in all generated test cases the depot can be operated 24 hours before the earliest time window start for any customer ship. This generates a need for the third aspect, which limits the earliest time a vessel can start its second voyage. The second voyage cannot be started before the vessel is finished with its first voyage, which again depends on the earliest time the vessel may finish operation at the first customer ship node on the first voyage. Necessary sailing times and the time of operating the depot must also be included in the expression of the earliest possible time for operating a customer ship node on the vessel's second voyage. Figure 13 shows a time line where this is illustrated. If $m > 1$, the following requirement holds for the time indices of the (v, i, m, t) combinations of the operating variables $y_{vimt} | i \in \mathcal{N}$:

$$
\begin{aligned}
t \geq \min_{j \in \mathcal{N} | j \neq i} \left(TW_j^{Start} + T_{vj}^{O} \right) + 2(m-1)T^{S} + (m-1)T_{v0}^{O} \\
+ \min_{j \in \mathcal{N} | j \neq i} (m-2)T_{vj}^{O}
\end{aligned}
\tag{5.8}
$$



A: $min\left(TW_j^{Start} + T_{vj}^{O} \right)$
B: Earliest time the depot can be operated by vessel v on voyage $m = 2$.
C: Earliest time node i can be operated by vessel v on voyage $m = 2$.
D: Earliest time the depot can be operated by vessel v on voyage $m = 3$.

Figure 13: Illustration of Aspect 3, showing a time line where the earliest time a vessel v can operate a node i or the depot on voyages 2 and 3 is indicated.

A similar expression to (5.8) exists for the depot. The time line of Figure 13 shows the earliest time the depot can be operated on the second and third voyages.

Combining the Aspects

Combining Aspects 1 and 2, we have the following lower bounds on the time indices of the (v, i, m, t) combinations of the operation variables $y_{vimt} | i \in \mathcal{N}$:

$$t \geq \max(TW_i^{Start}, T_v^E + mT_{v0}^O + (2m-1)T^S + \min_{j \in \mathcal{N} | j \neq i} (m-1)T_{vj}^O) \tag{5.9}$$

The time indices in the (v, i, m, t) combinations of the operating variables $y_{vimt} | (i \in \mathcal{N} \cup m > 1)$ must be greater than or equal to the maximum of expressions (5.8) and (5.9). Such combination of expressions can be made for the operating variables of the depot as well. With this, all three aspects are combined. Similar reasoning is used when defining the lower bounds of the time indices of the (v, i, j, m, t) and the (v, i, m, t) combinations of the sailing and waiting variables' subscripts, respectively. Algorithm 1, Algorithm 2 and Algorithm 3 give examples of how the operating, sailing and waiting variables are created.

In all test cases, the total quantity ordered from any customer ship does not exceed any vessel's total capacity. Hence, every vessel can operate all nodes at all customer ship. Because of this, it is not possible to decrease the number of possible (v, i) combinations for the operation, sailing or waiting variables.

Algorithm 1: Pseudo codes for creation of the sailing variables from the depot and between the customer nodes.

---

for all $v \in \mathcal{V}, i \in \mathcal{A}, j \in \mathcal{A}, m \in \mathcal{M}_v, t \in \mathcal{T} \mid i \neq j$ **do**

! Creating sailing variables from the depot

if $i = \{0\}$

and $i \neq j$

and $j \neq \{d\}$

and $t \leq \min(TW_0^{End}, TW_j^{End} - T_{vj}^O - T^S)$

and $t \geq T_v^E + mT_{v0}^O + 2(m-1)T^S + \min_{k \in \mathcal{N}}(m-1)T_{vk}^O$ **then**

if $m > 1$

and $t \geq \min_{k \in \mathcal{N}}\left(TW_k^{Start} + T_{vk}^O\right) + \left(1 + 2(m-2)\right)T^S$

$+(m-1)T_{v0}^O + \min_{(j \in \mathcal{N})}(m-2)T_{vj}^O$ **then**

create x_{vijmt}

else if $m = 1$ **then**

create x_{vijmt}

end-if

end if

! Creating sailing variables from customer nodes to other customer nodes

if $i \in \mathcal{N}$

and $j \in \mathcal{N}$

and $t \leq \min[\left(TW_i^{End}\right), \left(TW_j^{End} - T_{vj}^O - T^S\right)]$

and $t \geq \max[\left(TW_i^{Start} + T_{vi}^O\right),$

$\begin{pmatrix} T_v^E + mT_{v0}^O + (2m-1)T^S \\ + \min_{k \in \mathcal{N}}(m-1)T_{vk}^O + T_{vi}^O \end{pmatrix}]$ **then**

if $m > 1$

and $t \geq \min_{k \in \mathcal{N} \mid k \neq i}\left(TW_k^{Start} + T_{vk}^O\right) + 2(m-1)T^S + (m-1)T_{v0}^O +$

$\min_{(j \in \mathcal{N})}(m-2)T_{vj}^O$ **then**

create x_{vijmt}

else if $m = 1$ **then**

create x_{vijmt}

end if

end if

x_{vijmt} is binary

end do

Algorithm 2: Pseudo code for creation of the customer nodes' operating variables.

for all $v \in \mathcal{V}, i \in \mathcal{N} \cup \{0\}, m \in \mathcal{M}_v, t \in \mathcal{T}$ **do**

!Creating the operation variables of the customer nodes

if $i \in \mathcal{N}$

and $t \leq TW_i^{End} - T_{vi}^O$

and $t \geq \max[TW_i^{Start}, \begin{pmatrix} T_v^E + mT_{v0}^O + (2m-1)T^S \\ + \min_{j \in \mathcal{N}} (m-1)T_{vj}^O \end{pmatrix}]$ **then**

if $m > 1$

and $t \geq \min_{j \in \mathcal{N} | j \neq i} \left(TW_j^{Start} + T_{vj}^O\right) + 2(m-1)T^S + (m-1)T_{v0}^O + \min_{(j \in \mathcal{N})} (m-2)T_{vj}^O$ **then**

create y_{vimt}

else if $m = 1$ **then**

create y_{vimt}

end if

end if

y_{vimt} is binary

end do

Algorithm 3: Pseudo code for creation of the waiting variables for the depot.

for all $v \in \mathcal{V}, i \in \mathcal{A}, m \in \mathcal{M}_v, t \in \mathcal{T}$ **do**

! Creating waiting variables at the depot

if $i = \{0\}$

and $t < TW_0^{End} - T_{v0}^O$

and $t \geq \max[(T_v^E + mT_{v0}^O + 2mT^S + \min_{k \in \mathcal{N}} mT_{vk}^O),$
$(\min_{j \in \mathcal{N}} \left(TW_j^{Start} + T_{vj}^O\right) + (2m-1)T^S + (m-1)T_{v0}^O$
$+ \min_{(j \in \mathcal{N})} (m-1)T_{vj}^O)]$

create w_{vimt}

end if

end do

5.2.2 Creating the Remaining Variables

Almost all the other variables, both binary and continuous, are created with respect to the existence of either the sailing or the operation variables. As an example, it is not possible to have load on arcs that is not possible to sail. Hence; the arc-load flow variables, l_{vijfcm}, are only created for the (v, i, j, m) combinations where the sailing variables, x_{vijmt}, are exist. As mentioned, the complexity and size of the problem depend very much on the number of binary decision variables. Nevertheless, it is made an effort in reducing the number of all types of variables because the constraints are further created with respect to the existence of both the continuous and binary variables.

5.2.3 Creating the Constraints

As mentioned earlier in this chapter, constraints should only be created for relevant subscript combinations in order to avoid empty rows in the solver matrix. In the implemented model, each set of constraints is created with respect to the existence of the variables that are employed in the set, and the constraints are with this only created for relevant subscript combinations.

6 Valid Inequalities and Model Simplifications

This chapter presents three types of valid inequalities and two types of model simplifications. Adding the valid inequalities to the basic model in Chapter 4 may strengthen the model formulation, while model simplifications will make the model less complex. All valid inequalities and model simplifications are tested in order to see how they impact the model solving procedure. Results from these tests are given in Chapter 7. Section 6.1 presents the valid inequalities, while Section 6.2 presents the two types of model simplifications.

6.1 Valid Inequalities

Valid inequalities are linear inequalities which cut off the feasible region if added to the LP relaxation of an IP or MIP problem. This means that the optimal solution to the LP relaxation will provide a better optimistic bound for the solution to the IP or MIP problem. The valid inequalities should not alter the problem in any way, thus the optimal integer solution of the problem will remain the same.

6.1.1 Valid Inequalities Based on the LP Relaxation

The two types of valid inequalities presented in this section are developed by studying the optimal solutions of the LP-relaxed problem, and in this way finding connections between the variables.

In the MIP model, Constraints (6.1) ensure that a spot node i cannot be operated by a vessel v if the vessel is not utilized the day the node has its time window.

$$q_{vifm} - D_{if}\delta_{vt} \leq 0 \qquad \forall v \in \mathcal{V}, i \in \mathcal{N}_u, u \in \mathcal{U}^O, f \in \mathcal{F}, m \in \mathcal{M}_v, \; t \in \mathcal{T}^{DAY} | t \leq TW_u^{Start} < (t+H) \tag{6.1}$$

The q_{vifm} variables influence the income part of the objective function, and q_{vifm} variables will thus seek high values. In the linear relaxation of the MIP model, constraints (6.1) exploit the high values of the q_{vifm} variables to push up the δ_{vt} values. The δ_{vt} variables are originally binary variables stating whether a vessel is utilized a certain day. Since high values of the δ_{vt} variables reduce the objective function value, the linear relaxation has an incentive to reduce the value of the δ_{vt} variables, thus generating fractional values. Constraints (6.1) seek to reduce this incentive.

In the MIP model, constraints (6.2) below ensure that if vessel v' sails between nodes i and j on voyage m', then nodes i and j cannot be operated by other vessels than vessel v', or by vessel v' on other voyages than voyage m'. Constraints (6.2) coordinate the sailing and operation variables, and can be regarded as a case of clique inequalities on a given conflicting graph according to Agra et al. (2013).

$$\sum_{v\in\mathcal{V}\setminus v'}\sum_{m\in\mathcal{M}_v}\sum_{t\in\mathcal{T}} y_{vimt} + \sum_{v\in\mathcal{V}\setminus v'}\sum_{m\in\mathcal{M}_v}\sum_{t\in\mathcal{T}} y_{vjmt} + \sum_{m\in\mathcal{M}_{v'}\setminus m'}\sum_{t\in\mathcal{T}} y_{v'imt} + \sum_{m\in\mathcal{M}_{v'}\setminus m'}\sum_{t\in\mathcal{T}} y_{v'jmt} \leq 2\sum_{t\in\mathcal{T}}(1 - x_{v'ijm't}) \quad \forall v' \in \mathcal{V}, i,j \in \mathcal{N}, m' \in \mathcal{M}_{v'} \tag{6.2}$$

6.1.2 Cover Inequalities

Cover inequalities are problem specific valid inequalities, typically applied to problems with simple capacity constraints, like knapsack problems. A cover and minimal cover are defined in the following way by Lundgren et al. (2010):

If the feasible solutions to a 0/1 knapsack problem is given by the set $X = \{x \in \{0,1\}: \sum_{j\in S} a_j x_j \leq b\}$, *then the set* S *are a cover if* $\sum_{j\in S} a_j > b$. *The set* S *is also a minimal cover if for each selection of* $k \in S$, *we also have that* $S\setminus k$ *is not a cover, i.e.* $\sum_{j\in S} a_j - a_k \leq b$.

Further,

If S *is a minimal cover, then the constraint*

$$\sum_{j\in S} x_j \leq |S| - 1$$

is a valid inequality for X.

The cover inequalities to be added the basic mathematical model are based on the load constraints (4.17-4.25). Each vessel has an overall capacity, and the total loaded quantity on any voyage cannot exceed this upper limit. Because the problem includes multiple products and allocation to compartments, the cover inequalities are not as straight forward as described in the definition above. In our problem a given order quantity can be split between multiple compartments. Hence, it is not possible to make cover inequalities representing the capacity constraints for each compartment. Since the cover inequalities are defined with respect to the constraints' coefficients $a_{j|j \in S}$, they are very test case specific. More detailed description of how these inequalities are created and added to the basic model is therefore included in Chapter 7 after the test cases have been described.

6.2 Model Simplifications

The basic model presented in Chapter 4 has some complicating aspects which are likely to make the solution procedure more difficult. The model simplifications presented in this section can be included in the basic model individually or in combination to reduce the model's complexity.

6.2.1 Not Allowing Customer Splitting

The mathematical model described in Chapter 4 allows a customer ship to be operated by several supply vessels. Each node can only be operated by one vessel, but the different nodes at the same customer ship can be operated by different vessels. We call this aspect *customer splitting*. Customer splitting adds flexibility to the problem, which also increase the problem's complexity. In the real-life problem, customer splitting happens very seldom, meaning that almost every customer ship is only operated by one supply vessel. The model simplification presented in this section is forcing each customer ship to be operated by only one vessel by not allow customer splitting. When reducing the numbers of constraints and binary variables by doing this, the size and the complexity of the problem decrease.

With no customer splitting, some of the constraints from the original mathematical model presented in Chapter 4 must be replaced by two new sets of constraints. In addition, we eliminate the γ_{vum} variables which equal 1 if all nodes at customer ship u are operated by vessel v on voyage m. Constraints (4.11), (4.12) and (4.14) are eliminated, while constraints (4.13) are replaced with constraints (6.3) to still require continuous operation at every customer ship. In addition, constraints (6.4) are included to force the customer ships to be operated by one vessel each, if the customer ship is operated at all. Constraints (6.3) will force all nodes at the same customer ship to be operated on the same voyage. Hence, the operation variables of constraints (6.4) can be summed over the voyages $m \in \mathcal{M}_v$ to generate fewer constraints.

$$y_{vimt} - y_{v(i+1)m(t+T^O_{vi})} = 0 \qquad \forall v \in \mathcal{V}, i \in \mathcal{N}_u, u \in \mathcal{U}, m \in \mathcal{M}_v, t \in \mathcal{T} \tag{6.3}$$

$$\sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}_v} \sum_{j \in \mathcal{N}_u | j \neq i} y_{vjmt} - (|\mathcal{N}_u| - 1) \sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}_v} y_{vimt} = 0$$

$$\forall v \in \mathcal{V}, i \in \mathcal{N}_u, u \in \mathcal{U} \tag{6.4}$$

6.2.2 Eliminating Stowage

A ship routing and scheduling model with compartment allocation is a more complex problem than models with only simple capacity constraints. By eliminating stowage from the model, we therefore get a simplified problem. When solving the problem with simple capacity constraints, the allocation of fuels must be done manually afterwards to ensure that the solution is feasible.

Without stowage in the model, the model does not distinguish between the different fuel types and each vessel has only one single compartment. The capacity of this compartment will be the sum of the vessels' previous compartments' capacities, $\sum_{c \in \mathcal{C}_v} Q_{vc} = Q_v$. Without compartments and fuel types, the fuel allocation variables, k_{vfcm}, will no longer be relevant. The other types of load variables do no longer have subscripts of compartments and fuel type, meaning that the l_{vijm} variables replace the previous l_{vijfcm} variables and the q_{vim} variables replace the q_{vifm} variables. The quantities ordered by the customers will no longer specify fuel type, meaning D_{if} is replaced by D_i. When eliminating stowage, the load constraints (4.17) - (4.25) are replaced with the following constraints (6.5) - (6.11). Notice that constraints (4.23) and (4.24) are not included at all, while the others are reformulated.

$$\sum_{j \in \mathcal{N} \cup \{0\}} l_{vjim} - \sum_{t \in \mathcal{T}} D_i y_{vimt} - \sum_{j \in \mathcal{A}} l_{vijcm} = 0 \qquad \begin{array}{r} \forall v \in \mathcal{V}, i \in \mathcal{N}_u, \\ u \in \mathcal{U}^C, m \in \mathcal{M}_v \end{array} \tag{6.5}$$

$$\sum_{j \in \mathcal{N} \cup \{0\}} l_{vjim} - q_{vim} - \sum_{j \in \mathcal{A}} l_{vijm} = 0 \qquad \begin{array}{r} \forall v \in \mathcal{V}, i \in \mathcal{N}_u, \\ u \in \mathcal{U}^O, m \in \mathcal{M}_v \end{array} \tag{6.6}$$

$$\sum_{t\in\mathcal{T}} D_i^{MIN} y_{vimt} \leq q_{vim} \leq \sum_{t\in\mathcal{T}} D_i^{MAX} y_{vimt} \qquad \forall v \in \mathcal{V}, i \in \mathcal{N}_u, u \in \mathcal{U}^O, m \in \mathcal{M}_v \tag{6.7}$$

$$l_{vijm} - \sum_{t\in\mathcal{T}} Q_v x_{vijmt} \leq 0 \qquad \forall v \in \mathcal{V}, i \in \{0\}, j \in \mathcal{N}, m \in \mathcal{M}_v \tag{6.8}$$

$$l_{vijm} - \sum_{t\in\mathcal{T}} (Q_v - D_i) x_{vijmt} \leq 0 \qquad \forall v \in \mathcal{V}, i \in \mathcal{N}_u, j \in \mathcal{A} \; u \in \mathcal{U}^C, m \in \mathcal{M}_v \tag{6.9}$$

$$l_{vijm} - \sum_{t\in\mathcal{T}} (Q_v - D_i^{MIN}) x_{vijmt} \leq 0 \qquad \forall v \in \mathcal{V}, i \in \mathcal{N}_u, j \in \mathcal{A} \; u \in \mathcal{U}^O, m \in \mathcal{M}_v \tag{6.10}$$

$$\sum_{j\in\mathcal{N}} l_{vijm} = 0 \qquad \forall i \in \{0\} \cup \{d\}, v \in \mathcal{V}, m \in \mathcal{M}_v \tag{6.11}$$

7 Computational Studies

This chapter presents the results from testing the basic model presented in Chapter 4. We also look at results from the altered models which result from adding the valid inequalities and model simplifications presented in Chapter 6. The implemented models were run on a computer with specifications as stated in Chapter 5. Section 7.1 describes how the test cases are generated based on information from the Oil Company. Then we illustrate a specific solution in Section 7.2 in order to show how a test case solution may look like. Further, test results from the basic model are presented in Section 7.3, before Sections 7.4 and 7.5 present the test results from adding valid inequalities and model simplifications to the basic model. The problem's complexity and how the complexity and performance varies by altering the model will be discussed. The different models' ability to support the company in its decision making will also be analysed.

7.1 Generating the Test Cases

The test cases were generated based on data given by the Oil Company, mainly order lists from the autumn of 2011. These consist of a list of customer ships and their fuel orders specified by quantity and whether it is a fuel oil or gas oil order. Information regarding the vessels' daily fixed costs, average daily sailing costs, pumping rates, compartments and their load capacities, were also given. See Appendices A, B and C for more detailed information about the given data. In addition, pretesting during the implementation has made some guidelines in terms of size for the generation of the test cases.

Based on the assumptions of Chapter 2, the sailing times were set to one hour, independent of vessel, start point and destination. With an additional three hours in the cases of navy dock closure, the sailing times between the inner and outer port area were in these periods set to four hours. The sailing times between the customer nodes and the dummy end node were set to zero. With all these sailing times taken into account, the roughest discretization we can have without losing any information is a discretization of 24 time periods each day, where one time period represents one hour.

It was not given *which* type of fuel oil the different orders in the order list represented. As described in Chapter 2, about 80 - 85 % of the fuel oils ordered by the customers from the fuel supply companies are high sulphide fuel oils.

With this, 80 % of the fuel oil orders were set as high sulphide fuel oils, while the remaining 20 % were set as low sulphide fuel oils. The distribution between the two types of high sulphide fuel oil or the two types of low sulphide fuel oil were not given. Therefore, the two types of both high and low sulphide fuel oils were evenly distributed within their 80 % and 20 % part, respectively. With the mentioned requirements, the distribution of fuel oil types was generated by using the randomization function in Excel.

From the given order lists, the customer ships place between one and three different orders each. For every customer ship, the orders were sorted in a descending order with respect to the demanded quantity. This means that if a given customer ship is operated by only one vessel, constraints (4.13) ensure that the largest order is operated first, then the second largest, etc. As mentioned in Chapter 4, this predetermined order of operation was created in order to reduce symmetry.

For the spot customer ships, the demanded quantities given by the order lists were set as the upper bound of the delivered quantity, D_{if}^{MAX}. The lower bounds, D_{if}^{MIN}, were set to 90 % of the given demanded quantities. For contract customers, the quantity that the company must deliver is fixed. We chose to fix these quantities to the quantities which were given in the data from the company. Hence, the quantities demanded by contract customers are equal to D_{if}^{MAX} if the same customer ships are set as spot customers.

To make the test cases more similar to reality, some of the customer ships were assigned morning deliveries, meaning that they should be operated between 7:00 and 14:00 a specific day. Based on information from the Oil Company, about 20 % of the customer ships were assigned morning delivery, using the randomization function in Excel. For the customer ships with morning delivery, the start and end time parameters representing the customer ships' time windows were set to 7:00 and 14:00, respectively, on the specified delivery day. For all other customer ships their time windows were set to include all 24 hours of the specified day. It should be noticed that all randomized data have only been generated once. The same sets of randomized parameters have been used for all the test cases, in order to make comparison possible.

The operating times, T_{vi}^{O}, were for each vessel calculated by dividing the ordered quantities by the pumping rate of each vessel. In the real-life case, the pumping rate of the vessel and the customer ship must be compared, and the smaller rate will be the determining one. Since we had no information regarding pumping rates of customer ships, this simplified approach was utilized. Because of the model's discrete time approach, the operating time was rounded upwards. More specifically, since the model has discrete time periods of one hour, the operating time was rounded up to the nearest integer hour. As described in Chapter 2, the scheduler adds one third to the estimated operating time when scheduling. We assume that these thirds are, overall, included in the test cases by rounding up to the nearest integer hour. Consequently, we assume that times for coupling and decoupling pipes between the vessels and the customer ships are short enough to be included in this slack. This, in addition to the rounding up of the sailing times, is of course a simplified representation of reality, but it also makes the optimal solutions more robust in the real setting, as it adds some extra slack to the problem. For spot customer ships with flexible loads, the operating time is calculated based on the D_{if}^{MAX}-values. Based on the reasoning in Chapter 2, the operating time in the depot was set to 15 hours.

The revenue and cost parameters of the objective function, R_f, C_v^S and C_v^F, were scaled to ensure that it was always profitable to operate another spot customer ship, even if it includes one day extra of vessel usage and more sailing. Little information regarding the revenue was given from the Oil Company, so the revenue per operated quantity was set to be independent of fuel type. More specifically, we set $R = 1$, and scaled the cost parameters according to this. The cost parameters were estimated based on information from the Oil Company.

The number of time periods to include in the planning horizon was set to the end time parameter of the latest time window of the included customer ships; $|\mathcal{T}| = \max_{(i \in \mathcal{N})} TW_i^{End}$. The start of the planning horizon was set to $t = 0$. Starting with $t = 0$, this means that for instance time period $t = 4$ represents the hour between 4:00 and 5:00 of the first day, while time period $t = 28$ represents the same hour of the day on the second day. Vessel 1 was assigned to be available for loading at the depot from time period $t = 17$ meaning $T_1^E = 17$. Vessels 2 and 3 were set to be available for loading at the depot from $t = 7$ and $t = 0$, respectively. It is assumed that these values of the parameters T_v^E are representative for when the different vessels would be available a given day.

To avoid unreasonable long testing times, all tests have been run with a maximum running time of 10,000 seconds. This is also considered as a suitable amount of time for the Oil Company to make good schedules. Based on pretesting of the basic model during the implementation, the test cases have a maximum planning horizon of four days. With longer planning horizon feasible solutions were rarely found within 10,000 seconds. To avoid initial errors because of the long operating time in the depot, the earliest start of any customer ship's time window is 24 hours after the start of the planning horizon. This is illustrated in Figure 14.

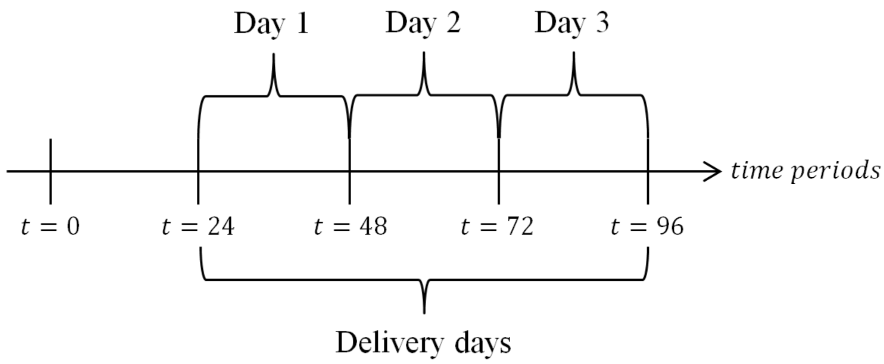


Figure 14: Time line showing the *delivery days*. All customer ships included in the test cases have time windows only within these days.

The pretesting during the implementation of the basic model also indicated that the test cases should not contain more than 12 customer ships distributed among the three *delivery days* illustrated in Figure 14. With a greater number of customer ships included, feasible solutions were rarely found within 10,000 seconds. To be able to observe the differences between the models when adding the various valid inequalities and model simplifications, the test cases must be of a certain degree of difficulty. Thus, the test cases generated include from 8 to 12 customer ships and are described in Table 2.

Table 2: The test case types where the numbers of ships of each delivery day is included in addition to the total number of customer ships and time periods of the various test case types.

Test Case Type	# Ships Day 1	# Ships Day 2	# Ships Day 3	# Ships in Total	# Time Periods
4_4_0	4	4	0	8	72
3_3_2	3	3	2	8	96
10_0_0	10	0	0	10	48
5_5_0	5	5	0	10	72
6_6_0	6	6	0	12	72
4_4_4	4	4	4	12	96

The mathematical model includes both spot and contract ships. To see how these different types of ships affect the solutions, it is chosen to have cases with either all ships as contract ships or all ships as spot ships. The situation with only contract ships is more realistic than having only spot ships. As described in Chapter 2, all mandatory orders in addition to the already confirmed orders are regarded as contract nodes, while new called in orders are regarded as spot nodes before they are accepted by the Oil Company. In the specifications of the test cases, *spot* is used if all ships are spot ships, while *contract* is used for cases with contract ships. *BM* indicates that the model tested is the basic model of Chapter 4. Hence, *BM_spot_10_0_0* represents a test case consisting of 10 spot customer ships the first delivery day solved by the basic model.

Even if the objective function (4.1) maximizes the profit where the revenue comes from delivered quantities to spot nodes, we do not change the objective function when testing on test cases with only contract nodes. Hence, the revenue in such cases is zero and the objective function will only comprise the daily fixed costs and the variable sailing costs. With this, the objective function value in test cases with only contract nodes will be negative and of a much smaller magnitude, since the magnitude of costs is much smaller than the magnitude of revenue. The gaps achieved within 10,000 seconds may then be greater for the contract cases, because the gaps then only represent the relative differences in costs.

7.2 Illustration of a Solution

The purpose of this section is to illustrate how a solution of a test case may look like. We will look at both routing, scheduling and the allocation part of the solution. The solution which is presented is from the test case BM_spot_3_3_2.

Table 3 contains information regarding the problem size and solution details. The best bound is the largest possible value the objective function may take, while the gap indicates how far off from the objective function value the best bound is. The presolved problem contains 29 % of the constraints and 82 % of the variables of the original problem in this case. The LP relaxation takes only 21 seconds to solve, while the MIP model still has a gap of 0.59 % after 10,000 seconds.

This test case consists of 14 orders placed by eight spot customer ships; three customers place orders on delivery day 1, three customers place orders on day 2, and two customers place orders on day 3. In the solution to this problem, all nodes are operated. Vessel 1 operates a total of eight nodes at four different customer ships on voyage 1. Vessel 3 operates three nodes at two customer ships on voyage 1, and three nodes at two customer ships on voyage 2. An illustration of the solution for vessel 3 is shown in Figure 15. We have chosen to illustrate only the solution for vessel 3 in order to make the illustration simpler.

Table 3: Problem size and solution details from the test case BM_spot_3_3_2. The problem size is given in numbers of constraints and variables. Solution details are given for the LP solution, the 1st MIP solution, the first MIP solution with gap below 10 % and the best solution after 10,000 seconds.

	Original Problem	Presolved Problem
Rows (constraints)	83836	24358
Columns (variables)	34975	28576
Integer Variables	17486	14987

	LP	1st MIP	Gap < 10 %	10,000 s
Objective Function Value	2547.49	287	2368	2483
Best Bound	-	2533.55	2533.55	2497.57
Gap	-	782.77 %	6.99 %	0.59 %
Solution Time	21	679	1119	-
Branch and Bound Nodes	-	0[1)]	240	12000

1) The solver found the solution by heuristics before it started with branch and bound.

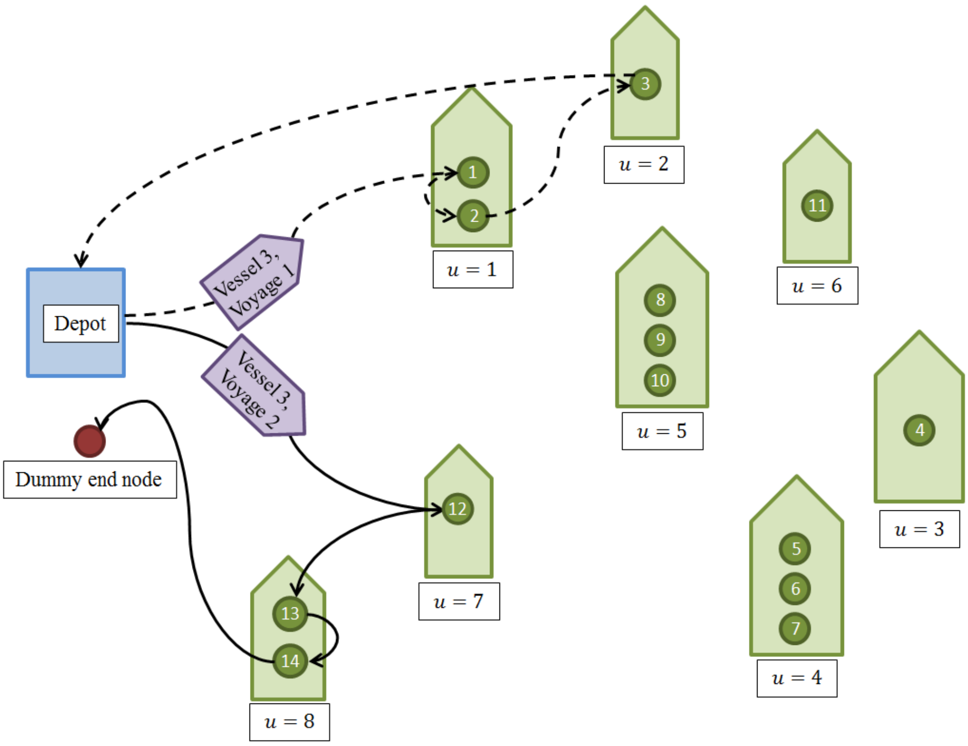


Figure 15: The solution of test case BM_spot_3_3_2 for vessel 3. Customer ships 3, 4, 5 and 6 are operated by vessel 1.

Figure 16 and Figure 17 show on time lines how vessel 3 executes its voyages. It starts by loading in the depot in time period 1. After 15 hours of operation, it sails from the depot to node 1. It arrives node 1 in time period 17, but since node 1 belongs to a customer ship with morning delivery, and hence has a time window which starts in time period 31, the vessel waits at node 1 until time period 32 before it starts operating the node. Directly after operating node 1 the vessel starts operating node 2, the other node at customer ship 1. The model is formulated such that there will be a fictive sailing between nodes belonging to the same customer ship. This sailing is omitted from Figure 16. After operating node 2, the vessel sails to node 3 on customer ship 2. It then starts operating node 3 in time period 35. After finishing operation at node 3, it sails back to the depot. Vessel 3 then waits at the depot from time period 38 until it starts operating in the depot in time period 72, as shown in Figure 17.

When vessel 3 starts operating in the depot this second time, it also starts its second voyage. Note that vessel 3 is waiting at the depot in all time periods 48 - 72. The vessel is thus not utilized at all the day these time periods represent, and the associated δ_{vt} variable is 0. This again means that there are no costs associated with using the vessel this day. This is in compliance with the objective function, which value would have been reduced if the vessel was utilized one time period or more this day. After loading in the depot, vessel 3 sails to node 12. It starts operating node 12 in time period 87, and starts sailing to node 13 in time period 89. It waits at node 13 in time period 90 before it starts operating in time period 91. After finishing node 13, it starts operating node 14, which belongs to the same customer ship. In time period 95, vessel 3 executes the fictive sailing from node 14 to the dummy end node.

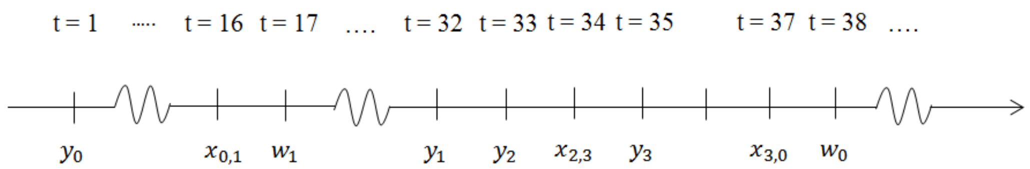


Figure 16: Time line of voyage 1 for vessel 3. y represents the time periods where the vessel starts operating a given node and x represents the time periods where the vessel starts sailing between two nodes. The time period where the vessel waits are indicated by w. All indices on the variables are node numbers. The broken line pieces indicate that time goes by without anything new happening. For instance, the vessel waits at node 1 in all time periods between the time periods 17 and 32.

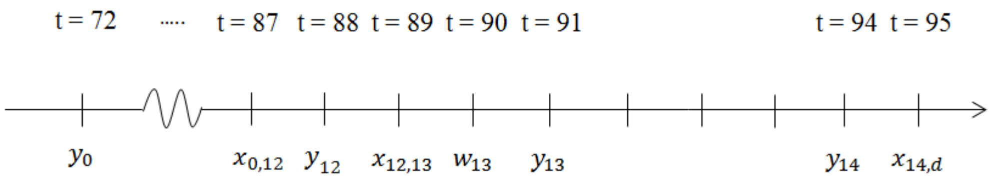


Figure 17: Time line of voyage 2 for vessel 3. See further explanations in the caption of Figure 16. Between the time periods 72 and 87 the vessel operates at the depot.

As already mentioned, vessel 1 operates the customer ships that vessel 3 does not operate. Vessel 1 starts operating in the depot in time period 26, and operates all its customers on one voyage. See Appendix D for a more detailed overview of the schedule of vessel 1.

This is an example of a solution where all customer ships with more than one node are served by one vessel only. Thus, the extra flexibility from customer splitting is not utilized. This also means that operation of the nodes at a customer ship happens continuously, as constraints (4.13) ensure this for all customer ships which are only operated by one vessel.

It should be noticed that there are many different ways which vessel 3's voyages could have been scheduled, assuming the customer ships it operates on each voyage are fixed. For instance, it could have operated customer ship 2 before customer ship 1. This could have been done since customer ship 2 does not have morning delivery; hence its time window starts in time period 24. For all customer ships which do not have morning deliveries, which for vessel 3 are all customer ships except ship 1, the time windows are 24 hours long. Since operation at a customer ship normally does not take very many time periods, there are many time periods where operation at a customer ship may begin, all of which would have given equally good solutions. There are also many ways the customer ships could have been allocated between the vessels. Hence, there is a lot of symmetry in the model. As stated in Chapter 4, constraints (4.13) ensure that if a vessel is to operate all nodes at a customer ship, it must operate the nodes in an ascending order. This reduces only some of the model's symmetry.

In order to get an impression of how the fuel allocation part of the problem is solved, we will look at how the fuels are allocated when vessel 3 leaves the depot on its second voyage. This is shown in Figure 18.

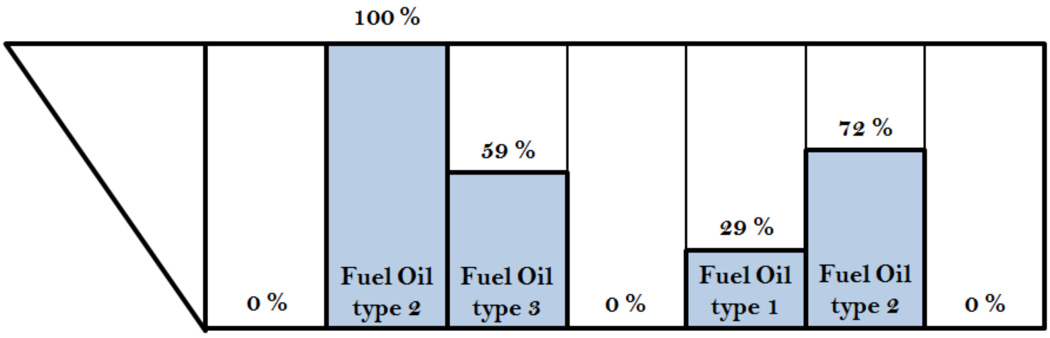


Figure 18: The load on board vessel 3 on voyage 2 when leaving the depot. Two of the compartments can only contain gas oil, the other five compartments can only contain fuel oils. Note that the compartments are not of equal size, even if the illustration may indicate this.

Vessel 3 has seven compartments, of which two are specified for gas oil, and five for fuel oils. On voyage 2, the gas oil compartments are empty, four of the fuel oil compartments contain a type of fuel oil while one fuel oil compartment is empty. In addition to indicating the fuel types the compartments carry, Figure 18 illustrates to what degree the compartment's capacities is utilized. As can be seen, the capacity utilization of the vessel is quite low. In total, 44 % of vessel 3's capacity is used on voyage 2. On voyage 1 the capacity utilization is even lower, only 20 %.

7.3 Test Results from Testing the Basic Model

The evaluation of test results from the basic model will focus on certain aspects. In accordance with the main objective of the thesis, see Chapter 1, we want to study the complexity of the problem. This will be done by evaluating the general performance of the model within the fixed time limit of 10,000 seconds. Further, we will assess the model's ability to support the company in its decision making. This will be done by looking at the best solutions the model is able to produce within the time limit, and how quickly it is able to obtain feasible solutions where all customers are operated. The latter is obtained in any MIP solution of a contract case. The same evaluation criteria will also be used later when evaluating valid inequalities and model simplifications.

The basic model from Chapter 4 is tested on the different test cases presented in Table 2. The results from testing the model on spot test cases are given in Table 4, while Table 5 presents the test results from testing on contract cases. Both tables present the objective function values, the best bounds and the gaps between these two values after 10,000 seconds. Table 5 includes the times to first

MIP solutions and their respective objective function values as well. Time to first MIP solution in spot cases is not included in Table 4 since only a very few customer ships are operated in the first MIP solution of spot cases.

Table 4: Test results from testing the basic mathematical model on test cases with spot nodes.

Test Case	After 10,000 seconds		
	Objective Function Value	Best Bound	Gap
BM_spot_4_4_0	2492	2513.00	0.84 %
BM_spot_3_3_2	2483	2497.57	0.58 %
BM_spot_10_0_0	2659	2989.35	12.42 %
BM_spot_5_5_0	2329	2989.88	28.38 %
BM_spot_6_6_0	1011	2490.41	245.24 %
BM_spot_4_4_4	2148	3485.98	62.29 %

Table 5: Test results from testing the basic mathematical model on test cases with contract nodes.

Test Case	After 10,000 seconds			Time to 1st MIP Solution [s]	Objective Function Value of 1st MIP
	Objective Function Value	Best Bound	Gap		
BM_ contract_4_4_0	-100	-49.06	50.94 %	880	-123
BM_ contract_3_3_2	-84	-62.97	25.04 %	908	-122
BM_ contract_10_0_0	-73	-47.64	35.02 %	9804	-77
BM_ contract_5_5_0	-105	-58.21	44.56 %	3673	-122
BM_ contract_6_6_0	No solution	-50.28	-	-	-
BM_ contract_4_4_4	No solution	-59.06	-	-	-

In the spot cases, the model finds a solution to all instances within 10,000 seconds. The gaps are below 1 % for the two cases with 8 customer ships; 3_3_2 and 4_4_0. In the cases of 12 spot customer ships, these gaps are more than 60 % and the objective values achieved after 10,000 seconds are in these cases probably far from the optimal solution. In the contract cases, the model is able to find a solution to four of the cases within the time limit. No solutions are found in the cases of 12 customer ships. In Chapter 2, it was stated that the scheduler must generate a feasible schedule including all customers within 10 minutes after a new spot customer's inquiry. Table 5 shows that the basic model is not able to find any MIP solutions in the contract cases within 10 minutes, or 600 seconds.

As described in Chapter 2, the Oil Company usually operates about six customer ships each day. As can be concluded from the results in this section, the basic model is too complex to solve the Oil Company's problem within a suitable amount of time. Due to long time windows (often 24 hours), short sailing distances (mostly one hour) and relatively short operating times (often one or two hours), there is a lot of symmetry in the model. There are also many ways to allocate fuels, which also add symmetry to the model. With this symmetry, there exists a great amount of possible solutions which are equally good, and which consequently make the solution procedure difficult.

7.4 Test Results from Adding Valid Inequalities

In Section 6.1 three types of valid inequalities was described. In this section, test results from adding some of these inequalities to the basic model are presented.

Some preliminary analyses indicate that cover inequalities will not significantly strengthen the basic model. The order quantities in the test cases are relatively small compared to the vessels' capacities; hence, numerous order quantities must be added in order to exceed the capacity limits. Because of this and the fact that there are multiple fuel types which cannot be loaded in the same compartment, only a small number of minimal covers can be generated. Based on the reasoning above it is chosen not to test the basic model with added cover inequalities. However, results from test cases including cover inequalities will be given for the simplified model in Section 7.5.2.

The two types of valid inequalities presented in Section 6.1.1 are added to the basic model in order to improve the best bound. In addition, these inequalities coordinate various variables, hopefully making it easier to achieve good feasible solutions. We will compare the results from the basic model presented in Section 7.3 with the results from the basic model with added valid inequalities. When evaluating the effects of the valid inequalities, the different LP solutions are compared to see whether the LP regions are reduced. The objective function values and gaps achieved after 10,000 seconds are also compared. The valid inequalities are tested individually and in combination. Table 6 shows an overview of the different tests executed with respect to test the effects of the valid inequalities.

Table 6: Overview of the tests of valid inequalities.

Test Denotation	Constraints Included
VI1	(6.1)
VI2	(6.2)
VI12	(6.1) and (6.2)

To test the effects of the valid inequalities, a representative sample of the test case types of Table 2 are tested. The chosen test case types are 10_0_0, 5_5_0 and 4_4_4. In Table 6, the test denotations VI1, VI2 and VI12 describe the valid inequalities added to the basic model in the various test cases. As an example, a test case with 10 spot customers the first day and constraints (6.1) and (6.2) added to the basic model will be denoted test case BM_VI12_spot_10_0_0. Constraints (6.1) include the flexible quantity variables q_{vifm}, which only exist for spot nodes. This implies that the valid inequalities VI1 will not affect the solutions in the contract cases. Thus, the contract cases are not tested with VI1 added to the basic model.

In Tables 7 and 8, the results from testing the basic model with and without added valid inequalities are presented. Table 7 presents the results for the spot test cases, while Table 8 presents the results for instances with only contract nodes.

Table 7: Testing the impact of the valid inequalities for the test instances with only spot nodes. The best results in each column; Objective Function Value, Best Bound, Gap and LP Solution for each test case type is marked in bold.

Test Case	After 10,000 seconds			LP Objective Value
	Objective Function Value	Best Bound	Gap	
BM_spot_10_0_0	2659	2989.35	12.42 %	3008.49
BM_VI1_spot_10_0_0	**2950**	2990.00	**1.36 %**	**3006.26**
BM_VI2_spot_10_0_0	2812	**2988.87**	6.32 %	3008.46
BM_VI12_spot_10_0_0	2840	2991.44	5.33 %	**3006.26**
BM_spot_5_5_0	**2329**	2989.88	**28.38 %**	3007.47
BM_VI1_spot_5_5_0	777	2987.17	284.45 %	**3001.73**
BM_VI2_spot_5_5_0	**2329**	2989.88	**28.38 %**	3007.47
BM_VI12_spot_5_5_0	944	**2987.00**	216.42 %	**3001.73**
BM_spot_4_4_4	2148	3485.98	62.29 %	3504.36
BM_VI1_spot_4_4_4	**3257**	**3475.80**	**6.72 %**	**3496.79**
BM_VI2_spot_4_4_4	2148	3485.98	62.29 %	3504.36
BM_VI12_spot_4_4_4	-	3476.57	-	**3496.79**

Table 8: Testing the impact of the valid inequalities for the test instances with only contract nodes. Note that valid inequality VI1 is not included, as it will not have any impact on the contract test cases. The best results in each column; Objective Function Value, Best Bound, Gap, Time to First MIP Solution and LP Solution for each test case type is marked in bold.

Test Case	After 10,000 seconds			Time to 1st MIP Sol [s]	Obj. Value of 1st MIP	LP Obj. Value
	Obj. Value	Best Bound	Gap			
BM_ contract_10_0_0	**-77**	**-46.33**	**39.83%**	**9804**	**-77**	**-29.54**
BM_VI2_ contract_10_0_0	No solution	-45.53	-	-	-	**-29.54**
BM_ contract_5_5_0	**-122**	-49.30	**59.59%**	**3673**	**-122**	-29.88
BM_VI2_ contract_5_5_0	-125	**-49.63**	60.29%	8532	**-125**	**-30.54**
BM_ contract_4_4_4	No solution	-59.06	-	-	-	-36.82
BM_VI2_ contract_4_4_4	No solution	**-60.97**	-	-	-	**-37.06**

Tables 7 and 8 show that the best LP bounds are always achieved by a model with some type of valid inequalities added, even if the improvements are not significant. Based on the results from Table 7 and 8, the plain basic model gives in total the best objective function values after 10,000 seconds. Nevertheless, by including valid inequalities VI1 better objective values and gaps are achieved in two of the three spot cases. Adding valid inequalities VI2 or the combination of inequalities VI12 does not seem to make any significant improvements, neither for spot nor contract cases. In the case contract_10_0_0, BM_VI2 is not able to find a feasible solution within 10,000 seconds, which the basic model is.

Based on the results from Table 7 and 8, it is difficult to state if the basic model in general perform best with or without valid inequalities VI1. To get a better basis for assessments, the remaining test case types from Table 2 are tested with and without these inequalities added. Since valid inequalities VI1 do not have any impact in contract cases, the tests have only been run as spot cases.

Table 9: Further testing of the impact of valid inequalities VI1.

Test Case	After 10,000 seconds			LP Solution
	Objective Function Value	Best Bound	Gap	
BM_spot_4_4_0	**2492**	**2513.00**	**0.84 %**	2546.59
BM_VI1_spot_4_4_0	**2492**	2514.18	0.89 %	**2539.36**
BM_spot_3_3_2	2483	**2497.57**	**0.58 %**	2547.49
BM_VI1_spot_3_3_2	**2490**	2505.13	0.61 %	**2536.38**
BM_spot_6_6_0	1011	**2490.41**	**146.33 %**	3506.26
BM_VI1_spot_6_6_0	**1032**	3488.33	238.03 %	**3501.14**

As for the previous tests, adding valid inequalities VI1 to the model gives better LP bounds than the basic model in the tests presented in Table 9. BM_VI1 also provides better solutions, though the differences are not significant. In spite of this, better best bounds and gaps are achieved by the plain basic model.

Based on the total result, it cannot be stated that adding valid inequalities improves the performance of the basic model. Hence, valid inequalities are not taken into consideration in further testing.

7.5 Test Results from Model Simplifications

In this section we will present results from tests on the model simplifications presented in Section 6.2. Based on the conclusion in the last section, valid inequalities are disregarded in these tests.

In order to evaluate how the model simplifications affect the performance, the results of the basic model with and without simplifications will be compared. It is important to notice that all models have the same objective function, and the objective values for the same test cases can thus be compared. When investigating how the model simplifications perform, we will focus on the same aspects as when testing the basic model, see Section 7.3.

7.5.1 Not Allowing Customer Splitting

In this section, we show results from the tests on the model simplification presented in Section 6.2.1. As explained in Section 6.2.1, we expect that removing the possibility of customer splitting will reduce the model's complexity. We have tested the simplified model on all test cases presented in Table 2, both as contract and as spot cases.

Table 10 presents the objective function value, the best bound and the gap after 10,000 seconds when testing on the cases with only spot customers. It also shows the time to optimal solution if an optimal solution is validated within the time limit. *NoCS* in the start of the test case name indicates that the model used is without customer splitting, otherwise is the intuition behind the test case names the same as in previous sections. Figure 19 graphs the best objective values after 10 000 seconds for the spot test cases. It compares the results from the BM model, given in Table 4, with the results from NoCS in Table 10.

Table 10: Results from testing without customer splitting on test cases with only spot customer ships.

Test Case	After 10,000 seconds			Time to Optimal Solution [s]
	Objective Function Value	Best Bound	Gap	
NoCS_spot_4_4_0	2503	2503.00	0.00 %	3669
NoCS_spot_3_3_2	2492	2496.17	0.17 %	-
NoCS_spot_10_0_0	2942	2987.05	1.53 %	-
NoCS_spot_5_5_0	2810	2982.38	6.13 %	-
NoCS_spot_6_6_0	3087	3485.04	12.89 %	-
NoCS_spot_4_4_4	2226	3477.13	56.26 %	-

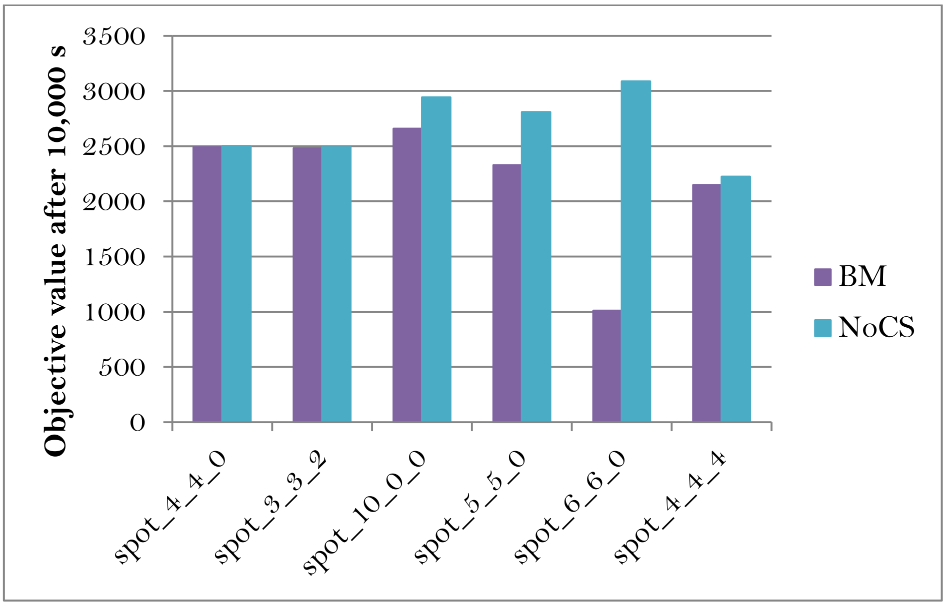


Figure 19: Comparing the objective values after 10,000 seconds for BM and NoCS.

Figure 19 shows that in all cases, NoCS produces equally good or better solutions than BM within 10,000 seconds. For test cases spot_4_4_0 and spot_3_3_2, the two models perform equally well. Of all the test cases, these are the ones comprising the least number of customer ships. In the larger test cases which are more difficult to solve, NoCS is able to come up with better solutions than BM. Hence, the value of having a simplified model is more noticeable in the test cases with a greater number of customer ships.

Table 11 presents the objective function value, the best bound and the gap after 10,000 second in the test cases with contract nodes. It also shows the time it takes to obtain a first MIP solution and time to optimal solution in the cases where an optimal solution is validated within the time limit. Note that NoCS is able to confirm contract_3_3_2 as optimal, but not spot_3_3_2, as shown in Table 10. Nevertheless, for 4_4_0 it takes longer time confirming the solution as optimal in the contract case, than in the spot case.

Table 11: Results from testing without customer splitting on test cases with only contract customer ships.

	After 10,000 seconds					
Test Case	**Obj. Func. Value**	**Best Bound**	**Gap**	**Time to Optimal Solution [s]**	**Time to 1st MIP Solution [s]**	**Obj. Func. Value of 1st MIP**
NoCS_ contract_4_4_0	-69	-69	0.00 %	8352	449	-87
NoCS_ contract_3_3_2	-80	-80	0.00 %	4589	618	-120
NoCS_ contract_10_0_0	-74	-53.75	27.37%	-	7361	-74
NoCS_ contract_5_5_0	-90	-59.87	33.48%	-	7588	-90
NoCS_ contract_6_6_0	No solution	-	-	-	-	-
NoCS_ contract_4_4_4	No solution	-	58.13%	-	-	-

Figure 20 graphs the time to finding first MIP solution in the contract cases. It compares BM, which results are given in Table 5, with the results from NoCS, as given in Table 11. Neither model finds a MIP solution within 10,000 seconds for test cases contract_6_6_0 and contract_4_4_4. Apart from these cases and contract_5_5_0, where BM performs better, NoCS produce MIP solutions more quickly. As can be seen from Table 11, in the two smaller cases NoCS finds a MIP solution within 10 minutes, something the basic model did not manage.

Still, the value of not allowing customer splitting is not as evident in the contract test cases as in the spot cases.

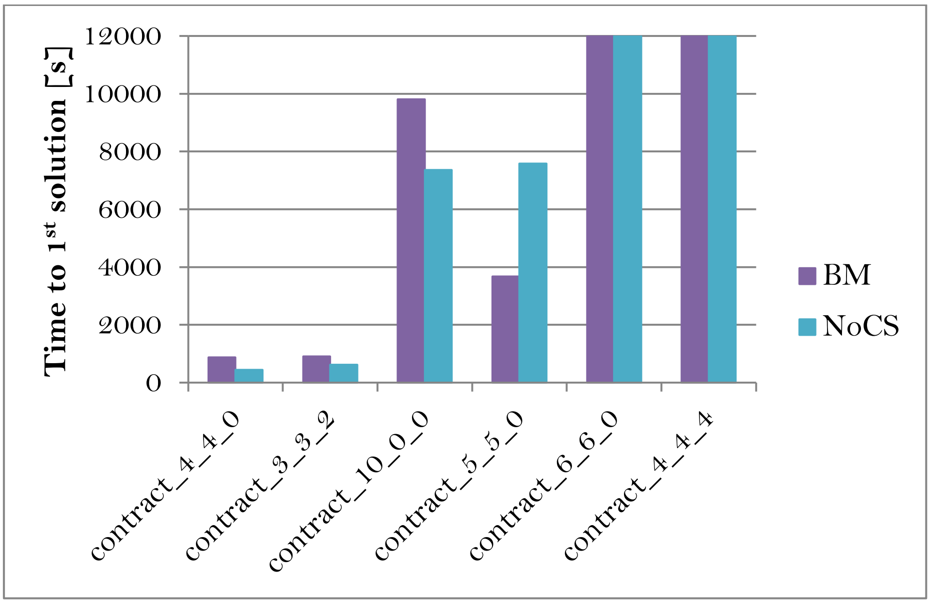


Figure 20: Comparing time to first MIP solution for BM and NoCS. Contract_6_6_0 and contract_4_4_4, for both models, have not found a solution within 10,000 s. The time to first MIP solution for these models is thus not known, and is therefore set infinitely high.

As a final comment, the test results show that a model where customer splitting is not allowed has a positive effect on the model's performance, especially in the test cases with spot customers.

7.5.2 Eliminating Stowage and Not Allowing Customer Splitting

Based on the promising results from the previous section, in this section we combine stowage elimination, as presented in Section 6.2.2, with not allowing customer splitting to test if this will further enhance the model's performance. With this, two complex aspects of the basic mathematical model described in Chapter 4 are removed.

As described in Section 6.2.2, the vessels have only got one single compartment each when eliminating stowage. Because of this, we risk getting solutions which are not feasible with respect to the real allocation problem. Hence, it is important to control that the obtained solutions are feasible by manually allocating fuels to compartments afterwards.

We have in the testing used different fractions of the actual total vessel capacity as vessel capacity, Q_v, in order to see how this affects the quality of the solutions. As running test cases with a low vessel capacity will increase the probability of getting a feasible solution, but may also force the vessels to operate fewer customers, there is a certain trade-off in choosing this fraction. We have tested the simplified model on all test cases presented in Table 2, both as contract and as spot cases. The different capacity fractions tested were 85 %, 70 %, 60 % and 50 %; where the latter fraction was the only one achieving solutions that always were feasible with respect to the real allocation requirements. The only results presented in this section are therefore from the tests with a capacity fraction of 50 %. This is indicated in the test case name, *ES50_NoCS*, describing this model with stowage elimination and no customer splitting. Table 12 presents the results from testing the model ES50_NoCS on the spot test cases.

Table 12: Test results from testing the ES50_NoCS-model with only spot nodes.

Test Case	After 10,000 seconds				Time to Optimal Solution [s]
	Objective Function Value	Best Bound	Gap	Feasible Solution?	
ES50_NoCS_ spot_4_4_0	2503	2503.00	0.00 %	Yes	247
ES50_NoCS_ spot_3_3_2	2492	2492.00	0.00 %	Yes	274
ES50_NoCS_ spot_10_0_0	2878	2879.50	0.05 %	Yes	-
ES50_NoCS_ spot_5_5_0	2945	2945.00	0.00 %	Yes	5500
ES50_NoCS_ spot_6_6_0	3434	3453.23	0.56 %	Yes	-
ES50_NoCS_ spot_4_4_4	3434	3434.00	0.00 %	Yes	7556

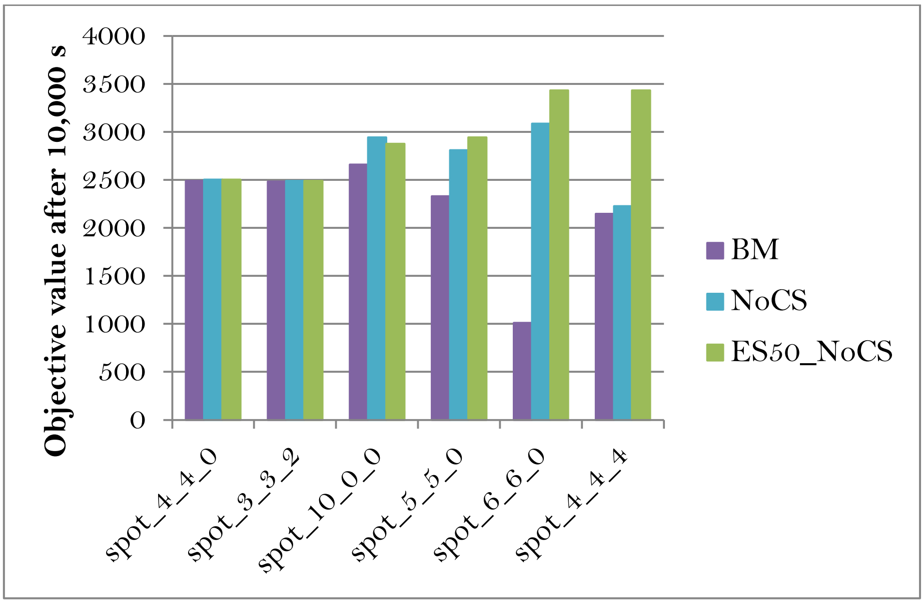


Figure 21: Comparing the objective values after 10,000 seconds for BM, NoCS and ES50_NoCS in cases of spot nodes.

Figure 21 graphs the objective values after 10,000 seconds in the spot test cases. It compares values obtained by BM, NoCS and ES50_NoCS. Results from BM and NoCS are given in Tables 4 and 10. ES50_NoCS obtains better objective value than the other models in all instances apart from test case spot_10_0_0, where NoCS performs better. After 10,000 seconds, ES50_NoCS has found a solution to this test case where 9 of 10 customers are operated, while NoCS was able to find a solution where all customers are operated. The best bound of 2879.50 obtained by ES50_NoCS is lower than the objective function of 2942 obtained with NoCS in this case. With this it can be stated that ES50_NoCS is unable to find a solution where all 10 customer ships are operated between time periods 24 and 48. The impossibility can be explained by the reduced vessel capacities combined with the long depot operating time. The latter forces the first day orders to mainly be operated by one vessel, which is difficult when the vessel capacities are only 50 %. Given the solution from ES50_NoCS, it may in practice be possible to allocate extra load to the vessels, thereby post optimizing the solution. Table 13 shows the results from testing the model ES50_NoCS on the contract test cases.

Table 13: Test results from testing the ES50_NoCS-model for the test instances with only contract nodes.

Test Case	After 10,000 seconds				Time to Opt. Sol. [s]	Time to 1st MIP Sol. [s]	Obj. Func. Value of 1st MIP
	Obj. Func. Value	Best Bound	Gap	Feas. Sol.?			
ES50_NoCS_ contract_4_4_0	-69	-69.00	0.00 %	Yes	304	65	-89
ES50_NoCS_ contract_3_3_2	-80	-80.00	0.00 %	Yes	274	122	-94
ES50_NoCS_ contract_10_0_0	No solution	-	-	-	-	-	-
ES50_NoCS_ contract_5_5_0	-93	-93.00	0.00 %	Yes	7577	270	-103
ES50_NoCS_ contract_6_6_0	-107	-87.10	18.60 %	Yes	-	1351	-118
ES50_NoCS_ contract_4_4_4	-107	-96.13	10.16 %	Yes	-	2443	-116

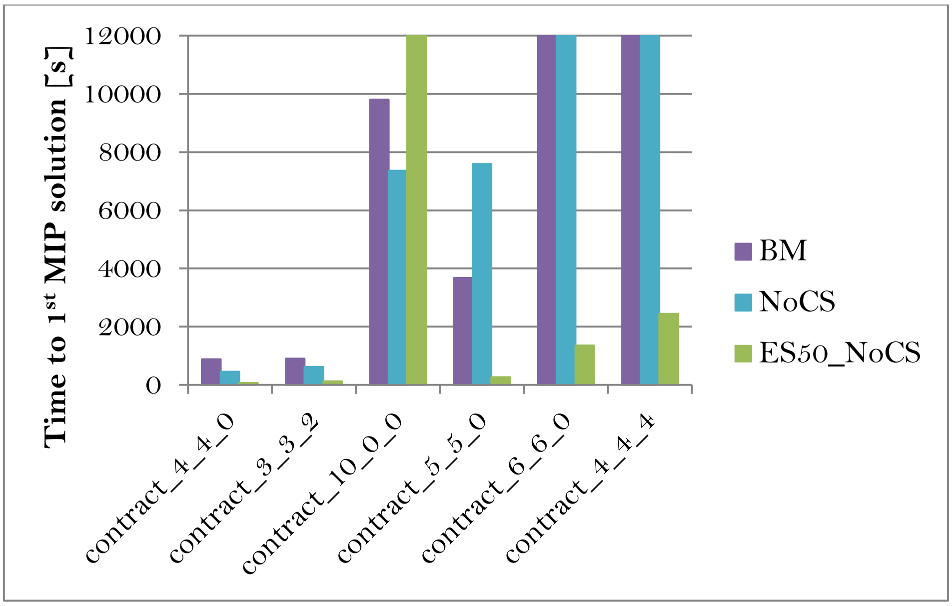


Figure 22: Comparing the times to first MIP solution for BM, NoCS and ES50_NoCS in cases of contract nodes. Note that infinitely high values are given for the cases where no MIP solutions are found within 10,000 seconds.

Figure 22 graphs the time to finding first MIP solution in test cases with only contract nodes. It compares results from BM, NoCS and ES50_NoCS. Results from BM and NoCS are given in Tables 5 and 11. In all test cases but one, MIP solutions are found in quicker manner with ES50_NoCS than with the other models. All these solutions are checked, and they were feasible with respect to the real allocation problem. ES50_NoCS is not able to find a MIP solution in the test instance contract_10_0_0. As shown in Appendix E, the solver confirms within 40 seconds that no MIP solutions exist. This confirms what was stated in the discussion concerning the results of the spot test cases; ES50_NoCS is not able to find solutions where all 10 customers are operated.

A reason to why the model with stowage elimination in general is able to perform better than the other models is the reduced problem size and complexity. Table 14 presents the number of generated rows and columns in the spot test cases for the different models.

Table 14: The number of rows and columns in the spot test cases, both for the original and the presolved problem. The number of rows represents the number of constraints, while the number of columns represents the variables.

Test Case	Original Problem		Presolved Problem	
	# Rows	# Columns	# Rows	# Columns
BM_spot_4_4_0	78372	30371	22627	25377
NoCS_spot_4_4_0	21472	30353	14991	22320
ES50_NoCS_spot_4_4_0	5144	15351	3042	10852
BM_spot_3_3_2	83836	34975	24358	28576
NoCS_spot_3_3_2	24668	34954	17777	27044
ES50_NoCS_spot_3_3_2	6487	18256	3564	13451
BM_spot_10_0_0	61349	45105	26422	36032
NoCS_spot_10_0_0	33801	45084	21481	29976
ES50_NoCS_spot_10_0_0	4592	17887	2554	12023
BM_spot_5_5_0	97575	46465	32886	39087
NoCS_spot_5_5_0	32865	46442	24467	36221
ES50_NoCS_spot_5_5_0	6545	22086	3946	16672
BM_spot_6_6_0	123792	62814	43674	53641
NoCS_spot_6_6_0	43905	62785	33812	50254
ES50_NoCS_spot_6_6_0	7938	29377	4838	22570
BM_spot_4_4_4	161184	67388	50092	58112
NoCS_spot_4_4_4	44005	67358	34129	55311
ES50_NoCS_spot_4_4_4	10131	35982	5812	27964

As can be seen from Table 14, the problem size is significantly reduced in the simplified models. Including both simplifications gives the smallest problem sizes. In the presolved problem the number of variables is reduced by 52-67 %, while the number of constraints is reduced by 85-90 %.

Testing Larger Test Cases

As the model ES50_NoCS performed very well on the test cases in Table 2, we have tried testing it on larger test instances. These test instances are described in Table 15 and are cases which better match real-life problem sizes. The cases presented in Table 15 are tested both as spot and as contract cases.

Table 15: Description of test cases which better match the real-life problem sizes.

Test Case Type	# Ships Day 1	# Ships Day 2	# Ships Day 3	# Ships in Total	# Time Periods
9_9_0	9	9	0	18	72
6_6_6	6	6	6	18	96
8_8_8	8	8	8	24	96

In the contract cases, feasible solutions were not obtained in any of the test cases. In test cases contract_8_8_8 and contract_9_9_0, it is confirmed after about 30 seconds that such solutions do not exist, see Appendix E. Due to the 50 % capacity fraction, it is possible that solutions for these cases exist, even though ES50_NoCS states it does not. As described earlier, this was the situation with test case contract_10_0_0. For test case contract_6_6_6, the solver was still working on finding the first feasible solution after 10,000 seconds.

Table 16 presents the test results from testing model ES50_NoCS on the larger spot test cases. Table 16 includes a column showing the number of ships operated in the best solution obtained within 10,000 seconds.

Table 16: Test results from testing ES50_NoCS on spot test cases of more real-life problem sizes.

Test Case	After 10,000 seconds				
	Objective Function Value	Best Bound	Gap	Feasible Solution?	# Ships Operated
ES50_NoCS_ spot_6_6_6	5260	5581.61	6.11%	Yes	15/18
ES50_NoCS_ spot_9_9_0	4895	5211.30	6.46%	Yes	12/18
ES50_NoCS_ spot_8_8_8	3869	6500.04	68.00%	Yes	6/24

Within 10,000 seconds, the gap of test case spot_8_8_8 is still 68 %. In the best solution, only 6 of 24 ships are operated, which implies that ES50_NoCS is not able to solve cases of this size satisfactorily. In spot_6_6_6 and spot_9_9_0, the gaps are of about 6 % after 10,000 seconds. In both cases, there are some customers which are not operated. The results from the contract cases show that ES50_NoCS states it is impossible to serve all customers maximum quantity in the cases 9_9_0 and 8_8_8. In the spot cases, the delivered quantities may be somewhat lower than in the contract cases. Still, it is unlikely that this flexibility is enough for ES50_NoCS to find solutions in spot_9_9_0 and spot_8_8_8 where all customers are operated, even with infinite running time.

Feasible Solutions within 10 Minutes
For the scheduler to answer the spot customers' inquiries there is a need for feasible solutions, or invalidating that such solutions exist, within 10 minutes. Based on the previous presented test results, we want to evaluate how the ES50_NoCS model performs with respect to this time limit. For both spot and contract cases, as shown in Table 12 and 13, optimal solutions were obtained within 10 minutes for the two smallest case instances. In addition, the first feasible MIP solution was obtained within 10 minutes for the case contract_5_5_0. Consequently, the 10 minutes limit is within reach for some of the smaller test cases. For two of the three larger test cases presented in Table 15, existing feasible MIP solutions were invalidated by ES50_NoCS within 30 seconds. Still, as earlier discussed, feasible MIP solutions may exist even if the model ES50_NoCS states it does not.

Testing Cover Inequalities
With the ES50_NoCS model, adding cover inequalities may be more effective than for the basic model. The total quantity to be delivered to the customer ships by a given vessel is now larger because all orders of a customer ship must be operated by the same vessel. In addition, by eliminating stowage cover inequalities can be generated simply based on the definitions from Section 6.1.2.

Since cover inequalities are very test case specific, it is chosen to only create covers for the two test cases of 12 customer ships; 6_6_0 and 4_4_4. For these two cases, minimal cover inequalities have been generated manually for both spot cases and contract cases. The covers for the spot cases are generated with respect to the lower quantity limits, D_{if}^{MIN}. For the contract cases, the covers are generated with respect to the fixed quantities, D_{if}. The denotation *CI* is used when the model is tested with cover inequalities added. Table 17 shows the results from testing ES50_NoCS_CI on the spot test cases, while Table 18 shows the results from testing on the contract cases.

Table 17: Test results from testing ES50_NoCS_CI on test cases with 12 spot customer ships.

Test Case	After 10,000 seconds			
	Objective Function Value	Best Bound	Gap	Feasible Solution?
ES50_NoCS_CI spot_6_6_0	3434	3445.63	0.34 %	Yes
ES50_NoCS_CI spot_4_4_4	3434	3439.45	0.16 %	Yes

Table 18: Test results from testing ES50_NoCS_CI on test cases with 12 contract customer ships.

Test Case	After 10,000 seconds				Time to 1st MIP Solution [s]	Obj. Func. Value of 1st MIP
	Obj. Func. Value	Best Bound	Gap	Feasible Solution?		
ES50_NoCS_CI contract_6_6_0	-107	-89.17	16.69 %	Yes	392	-107
ES50_NoCS_CI contract_4_4_4	-107	-97.61	8.78 %	Yes	561	-163

As given in Tables 17 and 18 above, all solutions achieved with ES50_NoCS_CI within 10,000 seconds are feasible with respect to the real allocation requirements. In the contract cases, the cover inequalities have a positive effect. The gaps obtained within 10,000 seconds are lower than with ES50_NoCS, which results are given in Table 13. Also, adding cover inequalities shortens the time to first MIP solution significantly. The objective function values obtained in all test cases are equal to the values obtained with ES50_NoCS without any added cover inequalities. For the spot cases, adding covers does not seem to have a significant effect. The gaps in the spot cases are in the same range as when cover inequalities are not included in the model. The spot quantity variables are maximized by the objective function, and they will therefore seek high values. As described, the minimal covers for the spot cases are based on the lower limit of the demand. Hence, these covers are most likely not very tight. This may be a reason to why adding the covers to the spot cases do not affect the solutions of the ES50_NoCS model significantly.

Figure 23 compares the time to the first MIP solution in contract test cases by ES50_NoCS and Es50_NoCS_CI. As already stated, adding the covers decreases the time to finding the first MIP solution. On average, the reduction is as large as 74 %. Hence, the cover inequalities improve the model significantly when testing on contract cases. The results therefore imply that minimal covers should be considered when ES50_NoCS are used to solve contract cases.

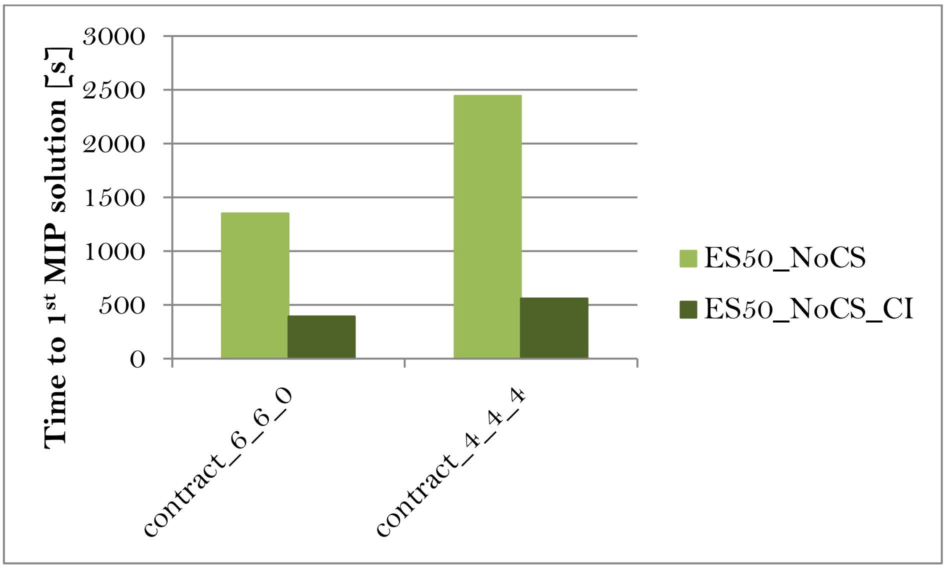


Figure 23: Comparing the time to first MIP solution for ES50_NoCS and ES50_NoCS_CI in the contract test cases.

Summarizing the Results from Testing ES50_NoCS

ES50_NoCS is the best performing model in this thesis. It finds better solutions faster than the other models, and in the contract cases it has shown to provide feasible solutions in a shorter time. Adding cover inequalities seems to improve the generation of feasible solutions in contract cases even more. Still, ES50_NoCS is not able to solve test cases of realistic sizes satisfactorily, so with regards to supporting the decision making, the results show that the model is not complete.

There is a downside of using this model, as the model may state that it is impossible to serve a given set of customers, even if it in reality is possible. This problem arises because of the low vessel capacities, which are necessary in order to get feasible solutions.

The fact that we had to reduce to a 50 % capacity fraction in order to generate feasible solutions implies that the allocation part of the problem is very difficult. In general, the customers' orders are often small compared to the size of the compartments, see Appendices B and C. In practice, it is possible for a small order to be the only order allocated to a compartment, which implies that a very little part of the compartment's capacity is utilized. When the ES50_NoCS model was tested with higher capacity fractions than 50 %, this was often a part of the cause to infeasible solutions. If the current compartments had been split into smaller ones, the allocation would probably have been easier, and the vessels could have utilized more of their total capacities. The company could therefore have been better off with a fleet of vessels with a higher number of smaller compartments in each vessel.

8 Concluding Remarks and Further Studies

In this thesis we have considered a combined fuel supply vessel scheduling and fuel type allocation problem, which was provided by a Hellenic oil company. The problem includes both contract and spot customers, where the contract customer's demands always have to be complied with. The spot customers must be answered within 10 minutes whether the company is able to serve them or not. Scheduling of the fuel supply vessels must comply with time windows of the customer ships, and should maximize the company's profit. There are multiple fuel types that the customers may demand, which must be allocated to separate compartments at the supply vessels.

The main objective of this thesis was to develop a detailed optimization model for the problem in order to really getting to know the problem and to study its complexity. Further, we wanted to utilize the model or variants of this to support the company in its decision making.

During the research three models have been evaluated; a basic model which comprises all relevant aspects of the real-life problem, and two models where some complicating aspects of the basic model are removed. One simplified model does not allow customer splitting, while the other include neither customer splitting nor stowage. The basic model was also tested with two types of valid inequalities added to strengthen the model.

All models have been implemented in Mosel mathematical programming language using the Xpress optimizer. The testing has been conducted on six test cases with three delivery days and a varying number of customer ships. The cases have been tested both with only spot customers and only contract customers. All six cases included fewer customers than a typical real-life case.

In the testing of the basic model, the model was able to find MIP solutions for all spot cases within the fixed time limit of 10,000 seconds. It was able to solve two of these cases with a gap below 1 %. In the contract cases, the model was able to find a MIP solution for four of the cases within the same time limit. The model was not able to find a feasible MIP solution within 10 minutes for any of the contract cases. Based on these results, it was concluded that the basic model is too complex for practical use.

The two types of valid inequalities were tested by adding them to the basic model and comparing the model's performance on the six test cases with and without the valid inequalities. Based on the results we concluded that adding these to the basic model did not improve the model significantly.

The two simplified models proved better than the basic model on the test cases. The best results were obtained by the model which includes neither stowage nor customer splitting. In the testing of this model, we used different fractions of the actual total vessel capacity as vessel capacity limit. We found that we had to reduce to a 50 % fraction in order to obtain solutions which were feasible with regards to allocation. This model solved all six spot test cases with a gap of 0.56 % or less within 10,000 seconds, four of which were solved to optimality. Two of these spot cases were solved to optimality within 10 minutes. In the contract cases, the model found a MIP solution within 10,000 seconds for five of the cases. A solution was not found for the sixth case, as the 50 % capacity limit was too constraining in this instance. A feasible MIP solution within 10 minutes was found for three of the contract cases, two of which were proven optimal within 10 minutes.

The model without stowage and customer splitting was also tested on three test cases of more realistic sizes. In the spot cases, the model solved two of the cases with a gap less than 7 % within the time limit. The model was not able to find feasible solutions in any of the contract cases within 10,000 seconds, and in the two largest cases it was confirmed quickly that no such solutions existed. It is probable that the 50 % capacity limit is a constraining factor in these test cases, which makes the results unreliable.

The fact that we had to reduce to a 50 % capacity fraction in order to generate feasible solutions in terms of allocation indicates that the allocation part of the problem is very difficult. If the current compartments had been split into smaller ones, the allocation would probably have been easier, and the vessels could have utilized more of their total capacities. Therefore, the company could have derived advantage from a fleet of vessels with a higher number of smaller compartments in each vessel.

Cover inequalities were developed for two contract cases and two spot cases, and applied to the model without stowage and customer splitting. The results showed that the covers generated for the contract cases had a large impact on how quickly the model was able to find a first MIP solution. The covers generated for the spot cases had little impact on the results.

Altogether, the main part of the thesis' objective is fulfilled by having formulated a detailed model of the real-life problem, and through this obtained a thorough understanding of the problem and its complexity. Further, we have seen that simplifying complicating model aspects reduces the model's complexity and makes the model better able to support the company in its decision making. Still, with regards to supporting the decision making, the results show that the model is not complete. In order to continue the improvements of the model's performance, there are a few areas that could be further investigated:

Heuristics have not been a part of our scope in this thesis. Still, at least for the part of the problem where feasible solutions are the important issue, it could certainly be an area of interest. For the company, it could be interesting to evaluate heuristics as a tool for generating solutions quickly, thereby supporting the decision making.

The current model has a time discretization of 24 hours per day, where each time period represents one hour. We have discussed the trade-off regarding the coarseness of the time discretization, as a finer discretization models the real problem in more detail while a coarser discretization reduces the problem size. It is most probably no point in making the time discretization finer, as this will further increase the problem size and most likely make the solving procedure even more difficult. On the other hand, making the discretization coarser will reduce the problem size, and may therefore be an interesting alternative. In that case, it must be investigated how a coarser discretization affect the modelling of the real life problem, as the level of detail probably will decrease.

An interesting alternative to our current model is to develop a path flow model, where feasible routes or voyages are pre-generated or dynamically created during the solving procedure. A master model will then choose the combination of routes or voyages which gives an optimal solution. If voyages are created in the sub problems, the master problem will have to include constraints which link the voyages together. If complete routes are generated in the sub problems, such constraints are not necessary, but the number of variables in the master problem will then increase. By including several of the problem's complicating aspects in the sub problems, a path flow model may perform better than the current arc flow model. Also, since the fuel allocation will be a part of the sub problems, we do not risk obtaining infeasible solutions, as in the model without allocation.

9 Bibliography

Agra, A., Andersson, H., Christiansen, M. and Wolsey L. (2012a). A maritime inventory routing problem: Discrete time formulation and valid inequalities. *Working Paper*, Norwegian University of Science and Technology, NTNU.

Agra, A., Christiansen, M. and Delgado, A. (2012b). Mixed integer formulations for a short sea fuel oil distribution problem. Transportation Science, INFORMS. (DOI: 10.1287/trsc.1120.0416)

Agra, A., Christiansen, M., Delgado, A. and Simonetti, L. (2013). Hybrid heuristics for a maritime short sea inventory routing problem. *Working Paper*, University of Aveiro, Aveiro, 1-31.

Andersson, A., Hoff, A., Christiansen, M., Hasle, G. and Løkketangen, A. (2010) Industrial aspects and literature survey: Combined inventory management and routing. Computers and Operations Research 37, 1515-1536.

Azi, N., Gendreau, M. and Potvin, J-Y. (2010). An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. European Journal of Operational Research 202, 756-763.

Al-Khayyal, F. and Hwang, S-J. (2007) Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, Part I: Applications and model. European Journal of Operational Research 176, 106-130.

Brønmo, G., Christiansen, M. and Nygreen, B. (2007). Ship routing and scheduling with flexible cargo sizes. Journal of Operational Research Society 58 (9), 1167-1177.

Brønmo, G., Nygreen, B. and Lysgaard, J. (2010). Column generation approaches to ship scheduling with flexible cargo sizes. European Journal of Operational Research 200 (1), 139-150.

Campbell, A. M. (2006). The vehicle routing problem with demand range. Annals of Operations Research (144)1, 99-110.

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. (2013). Ship routing and scheduling in a new millennium. European Journal of Operational Research 228 (3), 467-483.

Christiansen, M., Fagerholt, K., Nygreen, B. and Ronen, D. (2007). Maritime transportation. In Handbooks in Operational Research and Management Science, vol 14. Transportation, C. Barnhart and G. Laporte (eds.), (North-Holland, Amsterdam), 189-284.

Christiansen, M., Fagerholt, K. and Ronen, D. (2004). Ship routing and scheduling: status and perspectives. Transportation Science 38(1), 1-18.

Dauzère-Pérès, S., Nordli, A., Olstad, A., Haugen, K., Koester, U., Myrstad, P. O., Teistklub, G. and Reistad. A. (2007). Omya Hustadmarmor optimizes its supply chain for delivering calcium carbonate slurry to European paper manufacturers. Interfaces (1) 37, 39-51.

Donati, A. V., Montemanni, R., Casagrande, N., Rizzoli, A. E. and Gambardella, L. M. (2008). Time dependent vehicle routing problem with a multi ant colony system. European Journal of Operational Research 185, 1174-1191.

Erkut, E. and MacLean, D. (1992) Alberta's energy efficiency branch conducts transportation audits. Interfaces (22) 3, 15-21.

European Commission Eurostat (2013). http://epp.eurostat.ec.europa.eu/statisticsexplained/index.php/Maritime_ports_freight_and_passenger_statistics Last visited 14th of May 2013.

Fagerholt, K. and Christiansen, M. (2000). A combined ship scheduling and allocation problem. The Journal of the Operational Research Society (15) 7, 834-842.

Halvorsen-Weare, E. E. (2012). Maritime fleet planning and optimization under uncertainty. Doctor Theses. Norwegian University of Science and Technology, Trondheim, 55-58.

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.

Horgen, R. and Frich, O.-T. (2004). Supply chain optimization: Snøhvit LNG distribution. Master thesis. Norwegian University of Science and Technology, Trondheim.

Hvattum, L. M., Fagerholt, K. and Armentano, V. A. (2009). Tank allocation problems in maritime bulk shipping, Computers & Operations Research 36, 3051-3060.

Kobayashi, K. and Kubo, M. (2010). Optimization of oil tanker schedules by decomposition column generation, and time-space network techniques. Japan Journal of Industrial and Applied Mathematics, 27(1), 161-173.

Korsvik, J. E. and Fagerholt, K. (2010). A tabu search heuristic for ship routing and scheduling with flexible cargo quantities. Journal of Heuristics 16 (2), 117-137.

Lundgren, J., Rönnqvist, M. and Värbrand, P. (2010). Optimization. Student-litteratur AB, Lund, Sweden, 382-385.

Pang, K-W., Xu, Z. and Li, C-L. (2011). Ship routing problem with berthing time clash avoidance constraints. International Journal of Production Economics 131, 752-762.

Piraeus Port Authority S.A (2013). http://www.olp.gr/el/coastalshipping/coasting. Last visited 14th of May 2013.

Savelsbergh, M. and Song, J.-H. (2008). An optimization algorithm for the inventory routing problem with continuous moves. Computer & Operations Research (35) 7, 2266-2282.

UNCTAD (2012). Review of Maritime Transport 2012. United Nations, New York and Geneva.

Øvstebø, B. O., Hvattum, L. and Fagerholt, K. (2011). Routing and scheduling of RoRo ships with stowage constraints. Transportation Research Part C 19, 1225-1242.

10 Appendix

A. Calculating Maximum Loading Time

Based on the pump rates of the vessels and the refineries, the maximum loading time in the depot is calculated. At the refineries, gas oil and fuel oil can be loaded simultaneously. Since the total quantity of fuel oil is in general greater than the total quantity of gas oil, the vessels' fuel oil capacities will be determining when calculating the maximum loading time.

Table 19: Pumping rates and fuel oil capacity of the fuel supply vessels

Vessel	Pump Rate [m³/h]	Fuel Oil Capacity [m³]*
1	180	918
2	300	1559
3	320	2640

Table 20: Pumping rates of the refineries.

Refinery	Pump Rate [m³/h]
Elefsina	180
Aspropyrgos	280

The longest possible loading time occur when supply vessel 3 is fully loaded at Elefsina Refinery. The lowest pumping rate will be determining.

Maximum loading time = $\frac{2640\ m^3}{180\ m^3/h} = 14.66667\ h$.

B. Vessel Data

The tables of this appendix give information about the fixed supply vessel fleet. Note that the pump rates are given in Appendix A.

Table 21: The vessels' cost parameters

Vessel	Fixed Daily Costs	Variable Sailing Costs per Hour
1	12	1
2	15	2
3	17	2

Table 22: The vessels' capacity in total and on each compartment

Vessel	Capacity (Fuel Oil/Gas Oil)							
	Comp. 1	Comp. 2	Comp. 3	Comp. 4	Comp. 5	Comp. 6	Comp. 7	Total
1	226 (FO)	348 (GO)	372 (FO)	320 (FO)	40 (FO)	-	-	1306
2	228 (FO)	480 (FO)	360 (GO)	360 (FO)	335 (FO)	336 (FO)	70 (GO)	2189
3	380 (GO)	470 (FO)	510 (FO)	510 (FO)	510 (FO)	640 (FO)	97 (GO)	3117

C. Order Data

The same order list is used for all test cases. When having a test case with 12 customer ship, the 12 first listed ships of Table 21 are distributed in the given sequence among the delivery days. All ships are in the test cases either set to contract or spot ships. When all are contract ships, the quantity given as *Max Demand* in Table 21 are used as the fixed demanded quantity. The customer ships place up to four orders each. In the denotation of the fuel types, 5 represents gas oil, 1 and 2 high sulphide fuel oil and 3 and 4 low sulphide fuel oil. As seen, there are two fuel types of each of the fuel oils.

Table 23: Order list used in all test cases. The customer ships are given value 1 in the column if having morning delivery.

		Max. Demand/Min. Demand[m^3] (Fuel Type)			
Customer Ship	**Morning Delivery**	**Order 1**	**Order 2**	**Order 3**	**Order 4**
1	1	197/177(2)	37/33(3)		
2	0	400/360(2)			
3	0	320/288(2)			
4	1	91/82(1)	20/18(2)	9/8(5)	
5	0	45/41(1)	18/16(4)	5/5(5)	
6	0	50/45(2)			
7	0	300/270(3)			
8	0	930/837(2)	150/165(1)		
9	1	26/23(5)			
10	0	315/284(1)	85/77(2)	40/36(5)	
11	0	360/324(2)	117/105(3)		
12	1	26/23(2)			
13	0	17/15(1)			
14	0	91/82(3)	20/18(4)	9/8(5)	
15	0	288/259(1)			
16	1	181/163(1)	19/17(4)	9/8(5)	
17	0	960/864(1)	150/135(3)		
18	0	400/360(2)			
19	0	40/36(1)			
20	0	180/162(1)	40/36(2)		
21	0	299/269(2)	263/237(1)	38/34(4	27/24(5)
22	0	17/15(2)			
23	1	200/180(2)	100/90(1)		
24	0	400/360(1)			

D. Illustration of a Solution: Vessel 1's Schedule

Table 24: Vessel 1's schedule in test case BM_spot_3_3_2.

Time Period	Operating	Sailing	Waiting
25	y_{depot}	-	-
40	-	$x_{depot,4}$	-
41	y_4	-	-
43	-	$x_{4,5}$	-
44	-	-	w_5
45	-	-	w_5
46	-	-	w_5
47	-	-	w_5
48	-	-	w_5
49	-	-	w_5
50	-	-	w_5
51	-	-	w_5
52	-	-	w_5
53	-	-	w_5
54	-	-	w_5
55	-	-	w_5
56	-	-	w_5
57	y_5	-	-
58	y_6	-	-
59	y_7	-	-
60	-	$x_{7,11}$	-
61	-	-	w_{11}
62	y_{11}	-	-
63	-	$x_{11,8}$	-
64	y_8	-	-
65	y_9	-	-
66	y_{10}	-	-
67	-	$x_{10,dummy}$	-

If a variable equals 1 in a time period, then this is given in the variable's respective column. The indices on the variables indicate for operating and waiting variables which node vessel 1 is operating or waiting at, for the sailing variables they indicate which nodes the vessel sails between. Sailing between nodes at the same customer ship is omitted. Note that not all time periods between 25 and 67 are specified in the column with time periods.

E. Contract Cases with No Feasible Solutions

Even if no MIP solution is obtained within 10,000 seconds, such solution may exists if not the opposite is confirmed. Then the model is solves the problem to slowly to get MIP solutions. Note that ES50_NoCS is the only model tested on the three largest test cases. BM_VI1 are not tested on any contract case because valid inequalities VI1 do only affect the spot nodes.

Table 25: More details of contract cases where no MIP solution was found within 10,000 seconds. Note that the test cases of this table are only the test cases were no MIP solution was found within 10,000 seconds.

Test Case	Best Bound After 10,000 Seconds	Time to Confirm of No MIP Solution [s]
BM_VI2_contract_10_0_0	-45.53	-
ES50_NoCS_conctract_10_0_0	-87.84	40
BM_contract_6_6_0	-50.28	3099
NoCS_contract_6_6_0	-55.12	-
BM_contract_4_4_4	-59.06	-
BM_VI2_contract_4_4_4	-60.97	-
ES50_NoCS_contract_9_9_0	-53.30	14
ES50_NoCS_contract_6_6_6	-136.10	-
ES50_NoCS_contract_8_8_8	-43.62	28