NTNU
Norwegian University of
Science and Technology

Modeling of Seaborne Transport of Fresh Salmon

Inventory Routing with Continuous Time
Formulation for a Perishable Product

Marte Tuverud Kamphus

Marine Technology
Submission date: June 2017
Supervisor: Bjørn Egil Asbjørnslett, IMT
Co-supervisor: Inge Norstad, SINTEF Ocean AS

Norwegian University of Science and Technology
Department of Marine Technology

NTNU Trondheim
Norwegian University of Science and Technology
Department of Marine Technology

MASTER THESIS IN MARINE TECHNOLOGY

SPRING 2017

For stud.techn.

Marte Tuverud Kamphus

Modeling of Seaborne Transport of Fresh Salmon

Inventory Routing with Continuous Time Formulation for a Perishable Product

Background

Norway's production of farmed salmon is expected to grow from 1,31 million tonnes of farmed salmon in 2015 to 5 million tonnes in 2050. Every day, around 120 semi-trailers are on the roads, just with the mission of transporting fresh salmon from Norway to markets in Europe. The road network has an overall high load and the fish transport generates a big share. To handle future growth in the seafood industry it is therefore important to come up with new sustainable logistic solutions, and short sea shipping can be a solution to this problem.

Objective

The main objective of this thesis is to utilize optimization to model seaborne transport of fresh salmon from warehouse in Norway to warehouse in Europe. The model shall be used to gain insight into seaborne transport of fresh salmon.

Tasks

The candidate shall/is recommended to cover the following tasks in the master thesis:

a. Describe the real problem
b. Review and present relevant literature
c. Develop a mathematical model, which describes the simplified version of the real problem
d. Collect relevant data necessary for a computational study
e. Implement and solve the mathematical model in Xpress IVE
f. Use the results from the model to discuss problems regarding seaborne transportation of fresh salmon from Norway to Europe.

General

In the thesis the candidate shall present his personal contribution to the resolution of a problem within the scope of the thesis work.

Theories and conclusions should be based on a relevant methodological foundation that through mathematical derivations and/or logical reasoning identify the various steps in the deduction.

NTNU Trondheim
Norwegian University of Science and Technology
Department of Marine Technology

The candidate should utilize the existing possibilities for obtaining relevant literature.

The thesis should be organized in a rational manner to give a clear statement of assumptions, data, results, assessments, and conclusions. The text should be brief and to the point, with a clear language. Telegraphic language should be avoided.

The thesis shall contain the following elements: A text defining the scope, preface, list of contents, summary, main body of thesis, conclusions with recommendations for further work, list of symbols and acronyms, reference and (optional) appendices. All figures, tables and equations shall be numerated.

The supervisor may require that the candidate, in an early stage of the work, present a written plan for the completion of the work.
The original contribution of the candidate and material taken from other sources shall be clearly defined. Work from other sources shall be properly referenced using an acknowledged referencing system.

Deliverable
- The thesis shall be submitted in two (2) copies:
- Signed by the candidate
- The text defining the scope included
- In bound volume(s)
- Drawings and/or computer prints that cannot be bound should be organized in a separate folder.
- The bound volume shall be accompanied by a CD or DVD containing the written thesis in Word or PDF format. In case computer programs have been made as part of the thesis work, the source code shall be included. In case of experimental work, the experimental results shall be included in a suitable electronic format.

Supervision:
Main supervisor: Bjørn Egil Asbjørnslett

Sub-supervisor: Inge Norstad

Deadline: 18.06.2017

Preface

This Master Thesis is the last part of my Master of Science Degree at the Norwegian University of Science and Technology. The degree specialization is within Marine Design and Logistics at the Department of Marine Technology. The thesis is written in the 10^{th} semester, and the workload for the thesis is equivalent to 30.0 credits.

This Master is a continuation of my Project Thesis from the fall semester 2016, where I did a background study for seaborne transportation of fresh salmon from Norway to Europe. A simulation model was built to gain more system understanding. The background information for this Master Thesis therefore build up on the information that was obtained during the work with the Project Thesis.

I would like to express my gratitude to my supervisor Bjørn Egil Asbjørnslett, at the Department of Marine Technology for advice and support through the work with this Thesis. I would also like to express my gratitude to my sub-supervisor, Inge Norstad, at SINTEF Ocean AS for all his help regarding Mosel Xpress programming and problem discussion.

Trondheim, June 14, 2017

Marte Tuverud Kamphus

Summary

Norway produced 1,31 million tonnes of farmed salmon in 2015, and the production is expected to grow to 5 million tonnes in 2050. Every day, around 120 semi-trailers are on the roads, just with the mission of transporting fresh salmon from Norway to markets in Europe. The road network has an overall high load and the fish transport generates a big share. To handle future growth in the seafood industry it is therefore important to come up with new sustainable transportation solutions.

This report presents seaborne transportation of fresh salmon from Norway to Europe as a solution to future growth in the seafood industry. The aim of this thesis is to look at seaborne transportation and utilize optimization to build a mathematical model that transports salmon between loading and unloading ports, and to get insight and knowledge about the problem.

Transportation of seafood with semi-trailers are well utilized and the salmon gets fast to different markets. It will take a longer time to slaughter the salmon to fill up a ship, than filling up a semi-trailer. Fresh salmon is a perishable item and the durability is extremely important. The product degrades fast and it is important to track the time from the salmon first was slaughtered, to know exactly how old it is when it is delivered.

The biggest slaughter facilities in Norway often slaughter the salmon in three shifts per day. The production rate of salmon, that is assumed to be transported with ships, is therefore assumed to be constant per hour. The problem presented in this thesis is modeled as an Inventory Routing Problem, which enables the planners to evaluate both the inventory levels and the routing decisions. The mathematical model developed is a mixed integer model. Perishable considerations are included in the model to track the lead time and avoid waste of the product. To the authors knowledge perishability of items has never been explicitly modeled in maritime

inventory routing problems before.

The mathematical model is implemented in the commercial software FICO^{TM} Xpress Optimization Suit. It is tested on a case study for different combinations of loading and unloading ports. Result for one loading port and one unloading port with a production rate of 35 tonnes of salmon per hour gives a maximum lead time of 128 hours throughout the whole planning horizon. The problem is solved to optimality, and two ships are necessary in the solution. Increasing the production rate to 50 tonnes per hour, a solution with 10.7% optimality gap is obtained, and the model was stopped after 14 hours. The maximum lead time is decreased to 106.5 hours, but three vessels are necessary. The computational study for two ports shows that number of vessels necessary to keep the inventories satisfied and minimizing the lead time and cost are dependent on the production rate. The model has been tested with three ports as well, but few solutions have been obtained. A lot of time has been used on building a model with perishable considerations, and due to a developed model with high complexity fewer solutions than planned for have been obtained.

However, the model developed presents a new type of model that considers maritime inventory routing of a perishable asset. Tighter formulations are necessary for the model to solve the problem, but the model can be considered as a starting base for becoming a decision tool for the fish transportation in the future.

Sammendrag

Norge produserte i 2015 1,31 millioner tonn oppdrettslaks, og produksjonen er forventet å øke til 5 millioner tonn i 2050. Hver dag kjører rundt 120 trailere på veien med oppdrag å transportere fersk laks fra Norge til markeder i Europa. Veinettet har i dag høy belastning, og fisketransporten står for en stor del av denne andelen i Norge. For å håndtere fremtidig vekst i sjømatindustrien er det derfor viktig å komme med nye bærekraftige transportløsninger.

Denne rapporten presenterer sjøtransport av fersk laks fra Norge til Europa som en løsning på fremtidig vekst i sjømatindustrien. Formålet med oppgaven er å benytte optimering til å bygge en matematisk modell og bruke denne til å få mer innsikt og kunnskap om problemet.

Transport av sjømat med lastebiler er godt utnyttet og laksen blir transportert rask til de forskjellige markedene. Ved bruk av skip, må laksen vente lenger før den blir transportert enn ved bruk av lastebil. Fersk laks er et degraderbart produkt og holdbarheten er ekstremt viktig. Produktet nedbrytes raskt, og det er viktig å spore tiden siden laksen først ble slaktet, for å vite nøyaktig hvor gammel den er når den leveres.

De største slakteriene i Norge slakter ofte laksen i tre skift per dag. Produksjonsraten for laks, som antas å bli transportert med skip, er derfor antatt å være konstant per time. Problemet som presenteres i denne oppgaven, er modellert som et kombinert lagerstyrings- og ruteplanleggingsproblem, kalt Inventory Routing Problem (IRP), som gjør det mulig for planleggerne å evaluere både lagernivå og ruting av skipene. Den utviklede matematiske modellen er en blandet heltallsmodell. Siden laks er et degraderbart produkt, er tiden fra laksen først ble slaktet sporet gjennom hele modellen. Maritimt lagerstyrings- og ruteplanleggingsproblem med degraderbare produkter har til forfatterens kunnskap aldri blitt modellert før.

Denne modellen er derfor ett bidrag til dette forskningsområdet.

Den matematiske modellen er implementert i den kommersielle softwaren FICO™ Xpress Optimization Suit. Modellen er testet på et tenkt problem med forskjellige kombinasjoner av lastehavner og avlastningshavner. Resultat for en lastehavn og en avlastningshavn med produksjonsrate på 35 tonn laks i timen gir en maksimum ledetid på 128 timer gjennom planningsperioden. Optimal løsning er oppnådd, og to skip er nødvendig i den optimale løsningen. Økning av produksjonsraten til 50 tonn laks i timen, gir en løsning med 10.7% optimalitetsgap, etter at modellen var stoppet etter 14 timers kjøring. Maksimal ledetid er redusert til 106.5 timer og tre skip er nødvendig. Beregningsundersøkelsen for to havner viser at antall skip som er nødvendige for å opprettholde lagernivåene og minimere ledetid og kostnad er avhengig av produksjonshastigheten. Modellen er også testet med tre havner, men få løsninger er oppnådd. Mye tid har blitt brukt på å bygge en modell som tar hensyn til degraderbart produkt, og på grunn av en høy kompleksitet på den utviklede modeller, er færre løsninger enn planlagt blitt oppnådd.

Den utviklede modellen presenterer imidlertid en ny type modell innenfor Inventory Routing Problem med degraderbart produkt. Strengere formuleringer er derimot nødvendige for at modellen skal kunne løse problemet, men modellen kan betraktes som en startfase for å bli et avgjørelsesverktøy for fisketransporten i fremtiden.

Contents

List of Figures

List of Tables

List of Abbreviations

DWT Deadweight Tonnage

IRP Inventory Routing Problem

MAB Maximum Allowable Biomass

MIP Mixed Integer Problem

MIRP Maritime Inventory Routing Problem

m-PDP Multi-ship Pickup and Delivery Problem

SCM Supply Chain Management

VRP Vehicle Routing Problem

Chapter 1

Introduction

The Norwegian aquaculture industry started around 1970, and the industry has grown to become an industry of importance. Norway is today the world's leading producer of farmed salmon and the second largest seafood exporter in the world (The Norwegian Ministry of Trade and Fisheries, 2014). In 2015 Norway produced 1,31 million tonnes of farmed salmon with a value of 44,3 billions NOK (SSB, 2016). Figure 1.1 illustrates the growth in production since late nineties and the landed value of farmed salmon through the same period.

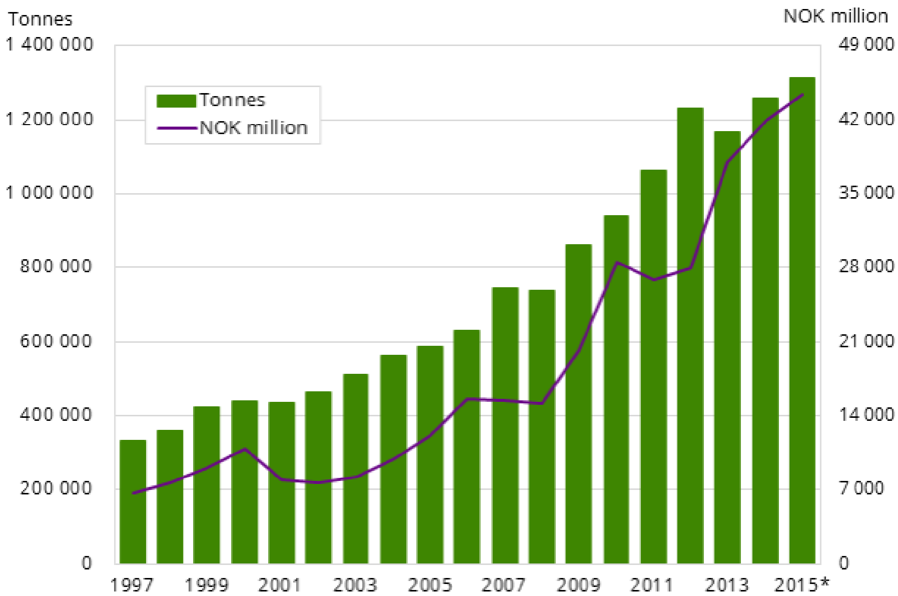

Figure 1.1: Produced amount and landed value of salmon. Source: (SSB, 2016)

The world's population is expected to grow past eight billion people by 2030 and ten billion people by 2050. The aquaculture industry will therefore play an important role for meeting the anticipated demand of food (Holmyard, 2016). The Norwegian aquaculture can contribute to both national and international food security, by increasing the production of farmed fish. The goal is to produce 5 million tonnes of farmed fish per year within 2050 (Laks, 2016). For this to be achievable the growth is dependent on sustainable solutions. Today around 80% of the salmon export is exported to Europe (SSB, 2016), and every day around 120 semi-trailers are on the roads, just with the mission of transporting fresh salmon from Norway to markets in Europe. The road network has an overall high load and Norwegian roads with many fjords and mountains might not be dimensioned for even higher loads than what the fish transport already generates today. To handle future growth in the seafood industry it is therefore important to come up with new sustainable logistic solutions, and short sea shipping can handle and are capable of accommodating these expected volumes (ECSA, 2016).

Today short sea shipping is not used to transport fresh salmon to Europe, but several ports in Mid-Norway have teamed up and are called the maritime gateway in Mid-Norway. They are working together on putting up a shipping route from Trøndelag to Europe and they have been working on this for the last ten years (Nord-Trøndelag Havn Rørvik, 2016). A solution to the problem might therefore be right around the corner, and this master thesis will look into seaborne transportation. The thesis is based on the work done in the project thesis, fall 2016. The project thesis contains a background study for seaborne transportation of fresh salmon from Norway to Europe. The objective of the project thesis was to look at today's transportation system and look at elements that have to be in place for fresh salmon to be transported to Europe with vessels.

The aim of the thesis is to utilize optimization to model a seaborne transport route between ports in Norway and Europe, with the main focus of not exceeding the salmon's shelf life. The mathematical model is structured as an Inventory Routing Problem (IRP), and will serve as a decision tool to get more insight into the possibilities for future short sea shipping of fresh salmon.

The thesis is structured as follows. Firstly, Chapter 2 gives an introduction to the salmon industry and how the transportation system is today. The chapter also explains some results from the project thesis. The methodology used to solve the problem and why choosing this methodology are described in Chapter 3. The problem description is described in Chapter 4, which is a simplification of the real problem. Relevant literature and research done on similar problems are presented in Chapter 5. The mathematical model developed for the simplified problem is explained in Chapter 6. The mathematical model is implemented in the commercial software FICO^{TM} Xpress Optimization Suit, and tested for different test cases. The implementation, test cases and the results are obtained in Chapter 7. Chapter 8 presents a discussion regarding the model and the problem, and at last a conclusion with further recommendations can be seen in Chapter 9.

Chapter 2

Background

This chapter provides background information that is considered relevant for this thesis. The salmon farming industry will be described from fertilization and hatching period to sales and distribution. The main focus will be on slaughter facilities and distribution. This chapter is based on the work done in the project thesis, but it is considered relevant as background for this master thesis.

2.1 Salmon Farming Industry

Supply Chain Management (SCM) is described by Chima (2011) as "the configuration, coordination and continuous improvement of a sequentially organized set of operations." All contributors in a distribution chain are highly dependent on each other. Which means that delays and unforeseen events may influence the remaining chain. The goal of a supply chain is to provide optimal service and try to maximize profit and minimize the cost along the way. When seaborne transportation constitutes one vital link in the chain, the supply chain is a maritime supply chain (Christiansen et al., 2007). The salmon farming industry contains breeding, stock fish, young fish production, salmon production, slaughter and processing and export and sales activities. An overview of the chain for food salmon is illustrated in Figure 2.1.

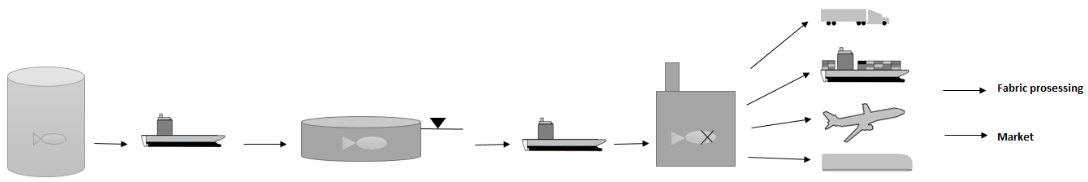

Figure 2.1: Overview of salmon farming

Production of salmon starts on land in freshwater tanks. The smolt is then transported out to sea. When the salmon is ready to be harvested, well-boats transport it to the slaughter facilities. When the salmon has been slaughtered it is distributed to markets or to further processing. Distribution of products outwards are called downstream supply chain, which is the focus in this thesis (Waters, 2003). Domestic distribution and export to Europe goes with semi-trailer. Export further away than Europe goes either with planes or ships. Today it is mostly frozen fish that is transported with ships (Hanssen et al., 2014). Different types of boats are well utilized in the production cycle of salmon, but not well utilized for distribution of slaughtered product, yet.

2.2 Production Cycle of Salmon

The following chapter explains the salmon production cycle more in detail. The information in this chapter is from Marine Harvest (2016). The cycle consists of two main phases, one in freshwater and one in seawater. The total production cycle lasts from 24 to 40 months. The first phase is in controlled freshwater environment. The broodstock is in saltwater, but the stripping and fertilization happens in freshwater. After egg hatching the alevins are feed through an attached yolk sac. They are referred to as fry when they have grown large enough to consume normal feed. When the freshwater phase goes towards the end they go through a smoltification process. This makes the fish ready for transfer to seawater. The smolt is transferred to the seawater net pens mainly twice a year, with well-boats. This is done to maintain a steady production throughout the year.

The second phase takes place in seawater. Each fish farm site usually has multiple net pens and can receive smolt in different batches. One net pen can hold a maximum of 200 000 fish (Høy et al., 2013). When the smolt is released into seawater the growth phase lasts for 14 to 24 months. The growth of salmon is strongly dependent on the sea temperature and the range for optimal temperature for Atlantic salmon is 8-14° C. The growth increases with temperature.

Fish farming companies are subject to regulations, and to do farming you need a license. In 2015, the number of licenses for Atlantic salmon and trout in seawater were limited to 974 licences. The production limitations are regulated as Maximum Allowable Biomass (MAB). This is defined as the maximum volume of fish a company can hold at sea at all times. One license is currently on 780 tonnes (945 tonnes in the counties of Troms and Finnmark), and a site generally holds between 2340 and 4680 allowed MAB.

When the salmon reach its target weight, around 5 kg, it is harvested and transferred to a processing facility with well-boats. The sizes of the well-boats vary between capacity of 100-700 tonnes of salmon. Throughout most of the year the harvesting volume is spread evenly, even though harvesting quantity is largest the last quarter of the year, since this is the period of best growth.

2.3 Slaughter Facilities

The slaughter facilities are mostly serviced with salmon evenly throughout the year. When the salmon arrive the facility, with well-boats, it is put into waiting cages. These cages also serve as resting cages. The salmon needs to rest for minimum a day before being slaughtered. Salmar's slaughter facility Innovamar has four waiting cages, which each has capacity of 350 tonnes salmon (Salmar, 2017). The facilities usually only slaughter and guts the fish, before it is put in boxes with ice, and transported for further processing or to the markets. The slaughter facilities are spread through the coast of Norway, as illustrated in Figure 2.2. The figure shows the location of the different facilities in 2015 and how much each county produced in the same (Fiskeridirektoratet, 2015). The figure was developed in the project thesis.

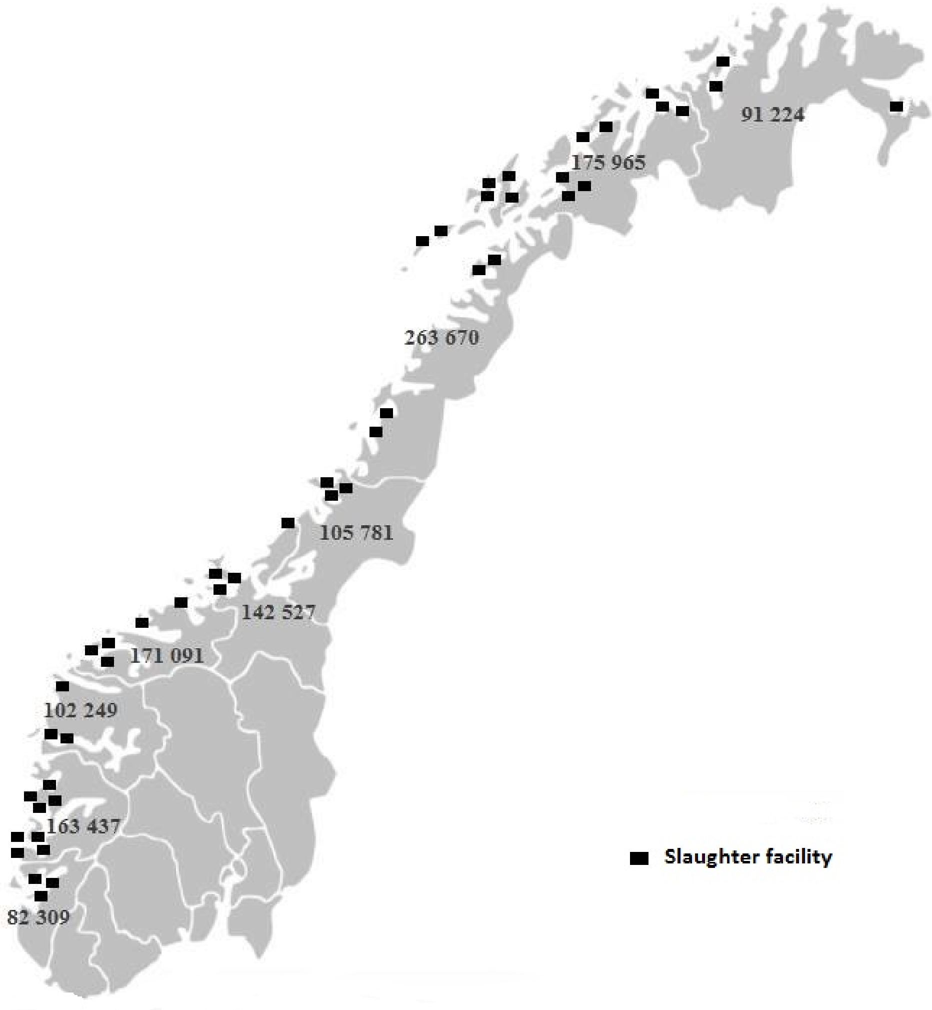

Figure 2.2: The points demonstrate the locations of the slaughter facilities. The numbers show the produced amount of salmon in each county in 2015

The last years, numbers of slaughter facilities have been reduced from 250 in 1986 to 47 today (Fiskeriøkonomisk and Norsk, 2013)(Hanssen et al., 2014) (Norwegian Food Safety Authority, 2016). This trend is expected to continue. It is considerable scale opportunities within aquaculture. Bigger netpens, bigger fish carrier and bigger slaughter facilities. To increase profit, the owners will use the opportunities they get, and it is therefore reasonably to assume that number of facilities will decrease.

The capacities of the slaughter facilities vary. Table 2.1 shows the capacity of three different processing facilities with their slaughter capacity per hour, per day and how many shifts they work per day (Farming company, 2016a).

Table 2.1: Slaughter capacity for three different facilities (Farming company, 2016a)

Slaughter facility	# fish per hour	# fish per day	# shift
Salmar Innovamar	7000-8000	130 000-180 000	3
Lerøy	2700-3000	50 000	2
Marine Harvest Ulvan	3700-43000	73 000	2

2.4 Distribution and Export

From the slaughter facilities the salmon is either transported away for further processing or directly to the markets and customers. Around 78% of the produced amount of fish was exported in 2015 (SSB, 2016). Where the salmon was exported can be seen in Table 2.2.

Table 2.2: The export to different countries and Continents in 2015 and the part of the total export

Country/Continent	Amount [tonnes]	Part of total export
Poland	139 435	13.5 %
France	121 033	11.7 %
Denmark	78 090	7.5 %
Great Britain	74 194	7.2 %
Spain	65 386	6.3 %
Netherlands	52 724	5.1 %
Rest of Europe	295 084	28.6 %
Total Europe	**825 946**	**79.9 %**
Africa	7479	0.7 %
Asia	155 582	15.1 %
North/Central America	41 176	4.0 %
South America	326	0.03 %
Oceania	2887	0.3 %
Total Export 2015	**1 033 396**	**100 %**

Table 2.2 shows the six largest importers in Europe for 2015 and the exported

amount to other Continents as well. From the table it can be seen that Poland is the biggest importer with 13.5% of the total exported amount in 2015. The table also shows that Europe is the overall biggest importer. Approximately 80% of the export goes to Europe.

Seasonal variations in export volumes can be seen in Figure 2.3. The graph shows the weekly exported value of fresh salmon throughout 2012, 2013, 2014, 2015 and 2016. The exported volumes trend to follow the same structure in these years. The weekly volumes varies with around 8000 tonnes, and overall more salmon are exported the last quarter. The fish are growing faster in these months and more fish are then brought to the slaughter facilities. Seasonal variations will be an important element when looking at a seaborne solution for fresh salmon. It is dimensioning for the maximal volume the vessel must be able to handle.

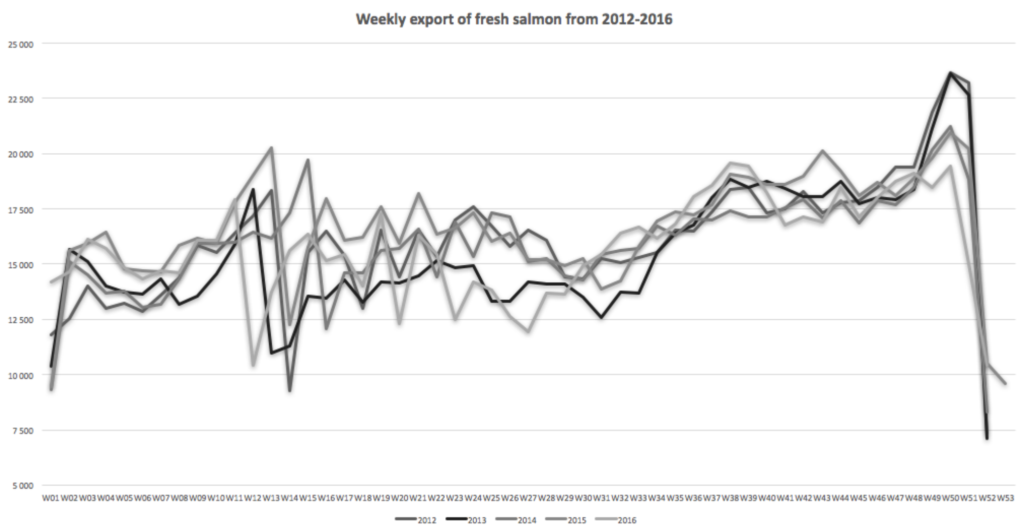

Figure 2.3: Weekly amount of fresh salmon in tonnes exported in 2012, 2013, 2014, 2015 and 2016

Figure 2.4 shows a chart of where the produced amount of salmon in 2015 ended up. It is split into five parts where domestic distribution is one, and export to the different Continents are the rest. It is assumed that the volumes that were not exported, were distributed domestic. On the left side you have how much salmon that were produced in 2015 and on the right where it ended up, how much and the percent compared to the produced amount can also be seen in the figure.

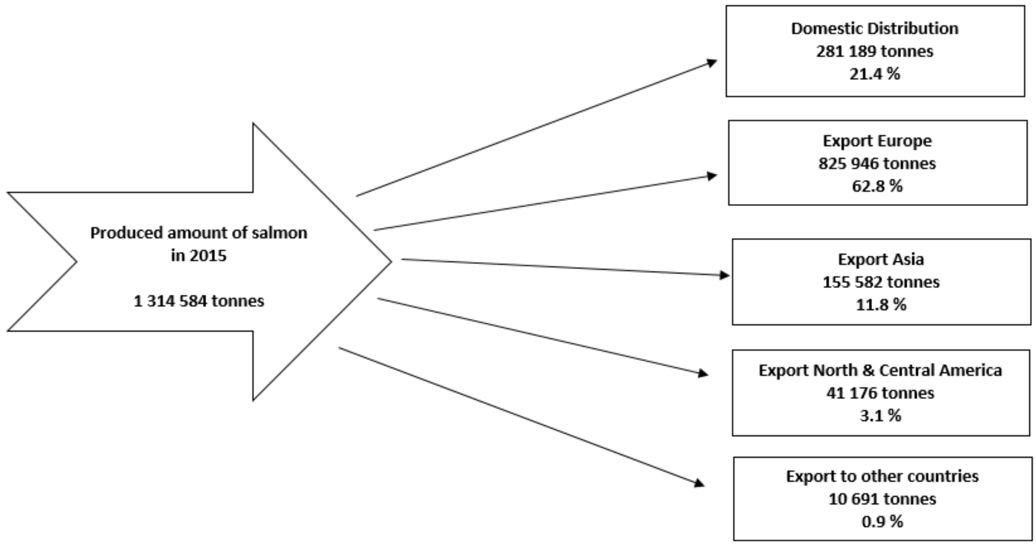

Figure 2.4: Where the produced salmon in 2015 ended up

Export to Europe is done with semi-trailers, and one vehicle can carry approximately 19 tonnes of salmon (Sinkaberghansen, 2016). The salmon is loaded into intermodal reefer containers designed and built for intermodal freight transport. The goods can then easily switch between different modes of transport and the cooling chain of the salmon can stay unbroken. Transportation time with semi-trailers from a slaughter facility on Hitra in Sør-Trøndelag to different locations at the Continent can be seen in Table 2.3. As shown in the table it will take a maximum of five days to transport the salmon from the slaughter facility to the customer.

Table 2.3: Transportation time from a slaughter facility to different customers at the Continental Europe (Farming company, 2016b)

From-To	Transportation time
Hitra -Northern-Europe	2.5-3 days
Hitra -Central-Europe	3-3.5 days
Hitra - Southern-Europe	3.5-5 days

2.5 Seaborne Transportation

Seaborne transportation of fresh salmon from Norway to Europe is not developed yet, but the Maritime Gateway in Mid-Norway are working on a shipping route for transportation of salmon from Rørvik and Hitra to Hirtshals in Denmark. On Hitra, port facilities for roll-on roll-off ships have already been established and in Rørvik a similar facility will be ready in 2017 (Nord-Trøndelag Havn Rørvik, 2016). Seaborne transportation of fresh seafood will not transport the load to the end-user, and it has to cooperate with other transportation modes. It is therefore important to have an efficient load system that makes it easy for further distribution at the Continent. Several load carriers are possible, but it is assumed the use of intermodal 45 fots reefer containers. These containers can easily be unloaded from a semi-trailer and loaded on a vessel, and be unloaded from the vessel and be loaded on the semi-trailer for further transportation.

A ship with size of 5000 deadweight tonnage (DWT) can transport the same load as 132 semi-trailers (Pedersen et al., 2006). Figure 2.5 illustrates this. If the slaughter facilities have the same capacities when slaughtering salmon for loading on semi-trailers as for a ship, the ship will need to wait before it can sail, while the semi-trailer can almost leave right away. Saying that generating salmon for one semi-trailer takes one hour, it will take five and a half day to generate the load for the ship. This means that the salmon will be much fresher when leaving with trailers, than a ship.

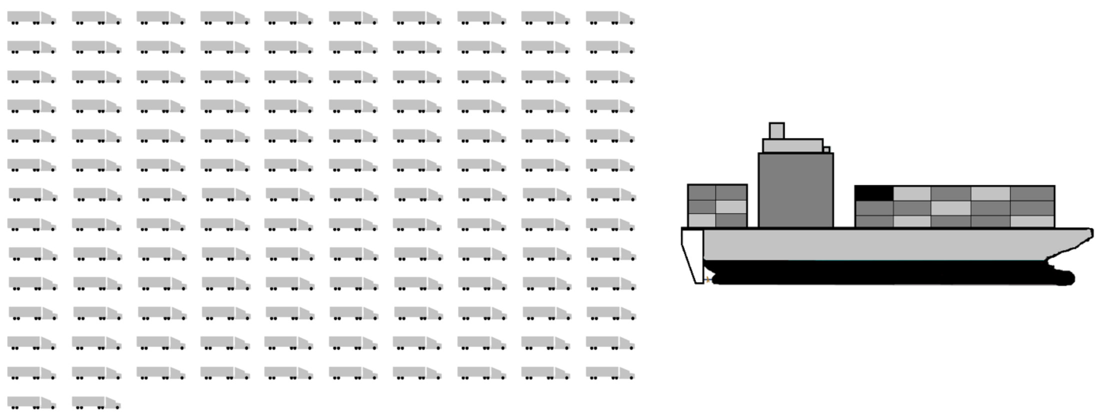

Figure 2.5: A ship with capacity for 2500 tonnes of salmon can carry the same amount as 132 semi-trailers

Perishable goods are sensitive to time and temperature (Dulebenets and Ozguven, 2017) and require specific conditions, independent on how they are transported. Norwegian Food Safety Authority has special rules for transportation temperature. Packed fresh fish products need to be cooled down to the temperature of the melting ice. It shall be around 0°C, and not exceed 2°C. When the products are in the store and waiting for the end-user, the temperature requirement is 4°C or lower (Mattilsynet, 2016). A perishable item is one that has constant utility up until an expiration date, at which point the utility drops to zero, after this point the product is no longer consumable (Nahmias and SpringerLink, 2011). Salmon is a perishable item and the temperature needs to be hold steady during the transport and the technology of the freight transportation needs to be able to do this. The shelf life of the fresh product is up to 3 weeks, and is defined from when slaughtering happens until consumption (Marine Harvest, 2016).

Applying superchilling technologies can increase the salmon's shelf life with one week. This process is defined as a method of preserving food by partial ice-crystallization. By applying this technology, the ice used for transportation is decreased. Superchilling does not need extra ice for transportation, and more salmon can be transported for the same weight. Superchilling can be an important technology for seaborne transportation, since a ship needs to wait a longer time than semi-trailers to be filled up. (Kaale, 2014)

In the project thesis, discrete event simulation, Matlab SimEvents, was used to gain more understanding about the system and the transportation route. Several routes were tested, and the project thesis focused on two ports in Norway that are potential for seaborne transportation. Nord-Trøndelag has three slaughterhouses and they produced 105 781 tonnes of salmon in 2015 (SSB, 2016). Rørvik Port in Nord-Trøndelag is located in between all the slaughter facilities and the distance to the facility furthest away is ten kilometers. Sør-Trøndelag has three large slaughter facilities, Lerøy Midt, Marine Harvest and Salmar and a smaller one Kråkøy. All together they produced 142 527 tonnes of salmon in 2015 (SSB, 2016). Hitra Kysthavn is placed at Jøsnøya and is under development. This port will be a natural hub for traffic and logistics. Several ports at the Continent are considered potential in the project thesis, but Cuxhaven in Germany and Gdynia in Poland have been used in the simulation model. Poland is the largest importer of salmon and further distribution from Gdynia can be done with semi-trailers.

The simulation model tested three different routes. Rørvik-Hitra-Cuxhaven-Rørvik, Rørvik-Hitra-Cuxhaven-Gdynia-Rørvik and Rørvik-Hitra-Gdynia-Rørvik. The model assumed that 70% of the total export to Europe goes to Cuxhaven, 87% goes to both distribution ports and 17% of the export goes to Gdyna. Export volumes were assumed equally divided over the counties. The model assumed a triangular distribution on the cargo generation rate, and was tested by increasing the cargo generation rate towards the 2050 production goal.

The results from the simulation showed that for the route to Cuxhaven it might be possible with seaborne transportation by using a vessel with capacity of 2500 tonnes salmon and a speed of 18 knots. The average time from starting port to approaching the delivery port will be 2.7 days. Throughout the test period, the storage in Rørvik is at times building up, so it might be a possibility to transport this with semi-trailers, to avoid that the salmon stay in stock for a longer period.

Simulation for delivery to two ports showed that the ships used too much time from the starting port to the last delivery port. It used from 4 to 5 days by doubling and tripling the export volumes, and that is without adding unloading time and transportation time to the end importer. Results for the route to Poland showed that if the volume is increased five times, it is still not advantageous to transport the fish at sea with vessels tested for capacity between 2300 and 3500 tonnes, and speed between 12-18 knots. It will take too long for the salmon to reach its end-importer. The sailing distance is too long and the generation rate is too low.

The simulation model does not take into account the time the salmon has been laying in stock before being loaded into a ship, which is an important aspect for a perishable item. This master thesis will therefore focus on the same problem as for the project thesis, but another method is used to gain even more understanding of the problem. Optimization will be used, and the focus will be on the salmon's shelf life. The problem will be modeled as an IRP.

Chapter 3

Methodology

The problem described in this thesis is solved with optimization. According to Lundgren et al. (2010) optimization is *"the science of making the best decision or making the best possible decision"*. The first applications to optimization were military and goes back to World War II (Lundgren et al., 2010), but today optimization is used in a large number of economic and technical applications. The models are used in both operative planning and tactical and strategic planning. Optimization is used to provide support for decisions in a real problem, by simulating the problem and by testing several scenarios and evaluate cause and effect when input data are changed.

This chapter will provide basic information on the optimization process and explain the mathematical model that is used to solve the problem in this thesis. This chapter is not meant to give the reader a full explanation of the mathematical model presented in Chapter 6, but to give the reader additional support to better understand the mathematical model.

3.1 Optimization Process

A special working approach is used when an optimization model is used to analyze and solve a given problem. This is considered as an optimization process and contains 4 phases; *identify, formulate, solve* and *evaluate*. These phases are often performed in parallel and the completion time depends on the size, complexity, structure and properties of the problem. An overview of the process according to Lundgren et al. (2010) is given in Figure 3.1.

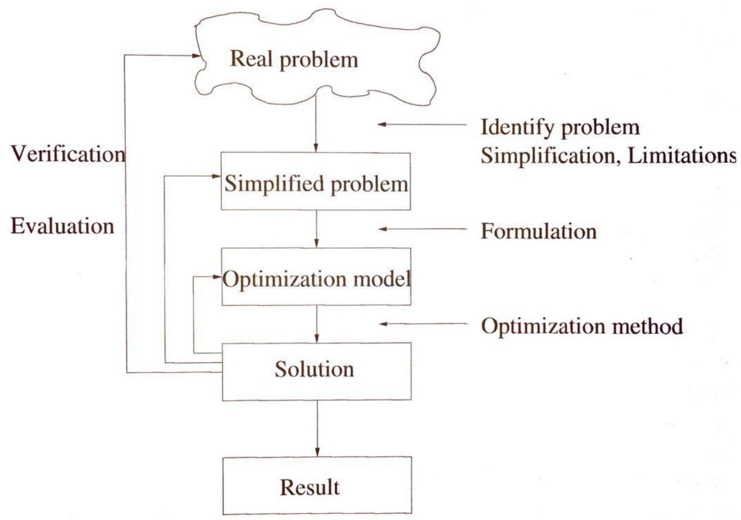

Figure 3.1: The optimization process. Source: (Lundgren et al., 2010)

The real problem represents the problem that needs to be solved. The real problem in this thesis is presented in Chapter 2. The problem is often complex and several elements cannot be included in an optimization model. It is therefore important to *identify* elements that are irrelevant or not important, and the remaining problem is referred to as the simplified problem, which in this thesis is provided in Chapter 4.

The next step is then to *formulate* the simplified problem as an optimization model, mathematically. The problem now contains decision variables, an objective function and constraints. How solvable the model is depends on the model's structure and problem size. Several simplifications may be necessary to make. The problem also depends on the amount of data available, and how reliable they are.

After the optimization model is made it must be *solved.* This is often done by solution algorithms or implementation into commercial software. This thesis uses a commercial software named FICO™ Xpress. To be able to solve the model, the data must be gathered. Collection of correct data to model the real problem can be challenging.

After the model is solved the last step is to *evaluate* and verify the solution. This phase should check that the solution is correct based on the mathematical formulated problem and that the model describes the problem accurately enough.

Sensitivity analyses can be used to determine the effect with changed input data. When the solution is validated the mathematical model can be used as a support tool for decisions in the actual problem.

3.2 Mathematical Modeling for This Thesis

The problem in this master thesis is modeled as an IRP. The reason for choosing this methodology is that it is wanted that the product controls the routing and scheduling of the vessels. The model utilizes the same visiting system and time formulation as Christiansen et al. (2007). The visiting system enumerates each visit at a node in the network, and treats the time as continuous. By doing this the time horizon does not have to be divided into periods. The product is produced at loading ports, and consumed at unloading ports. Inventory storage capacities are given in all ports. When modeling the problem as IRP the number of visits at each port during the planning period are not predetermined, nor is the quantity to be loaded or unloaded in each port. How the visiting system works is illustrated in Figure 3.2.

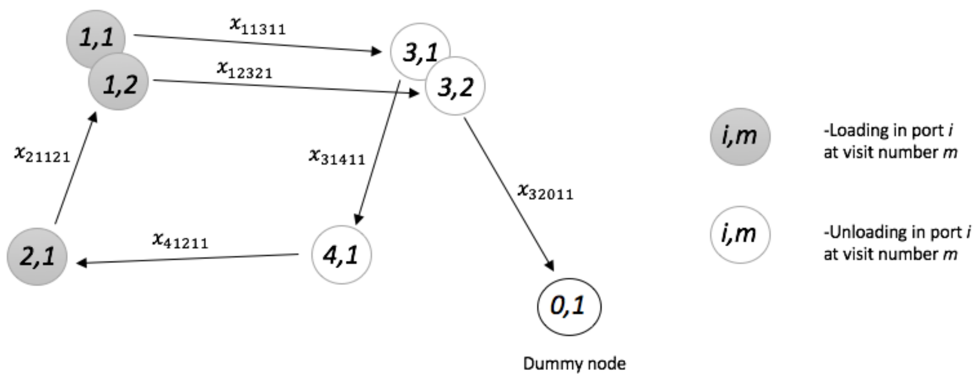

Figure 3.2: Visiting system for two loading ports and two unloading ports

Figure 3.2 illustrates a sailing pattern for one vessel. The variable x_{imjnv} shows the vessel's movement in the network, between the port calls *(i,m)*. The vessel starts in loading port 1, where it loads, before it sails to unload in port 3 and port 4. From there the vessel sails to load in port 2 and 1 again, before it sails to unload in port 3, before leaving the network and ends in the dummy end node. The inventory level at a loading port is illustrated in Figure 3.3. The production rate is constant per time unit and the inventory level can only be reduced if a vessel pickup load.

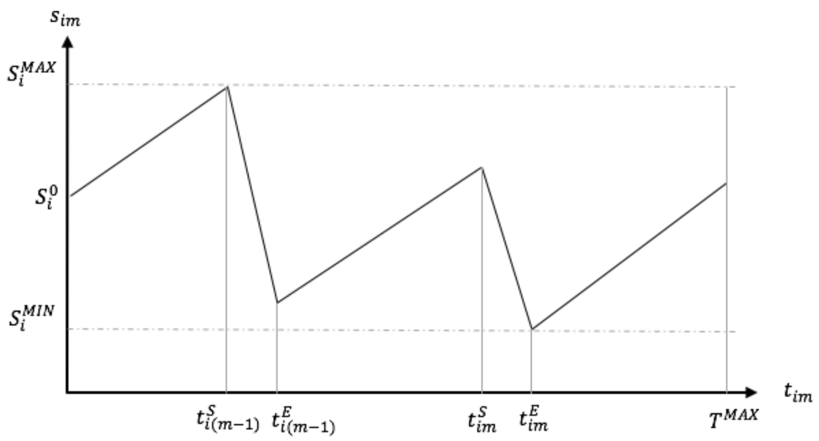

Figure 3.3: Example of inventory level during the planning horizon for a loading port

The inventory level at an unloading port is illustrated in Figure 3.4. Same as for the loading port, the consumption rate is constant, and the inventory level can only increase if there is a delivery. To avoid shortage of the product a lower safety stock level that is above a specified minimum storage capacity is defined. Any levels below the lower safety stock level, will be penalized, as illustrated in the figure.

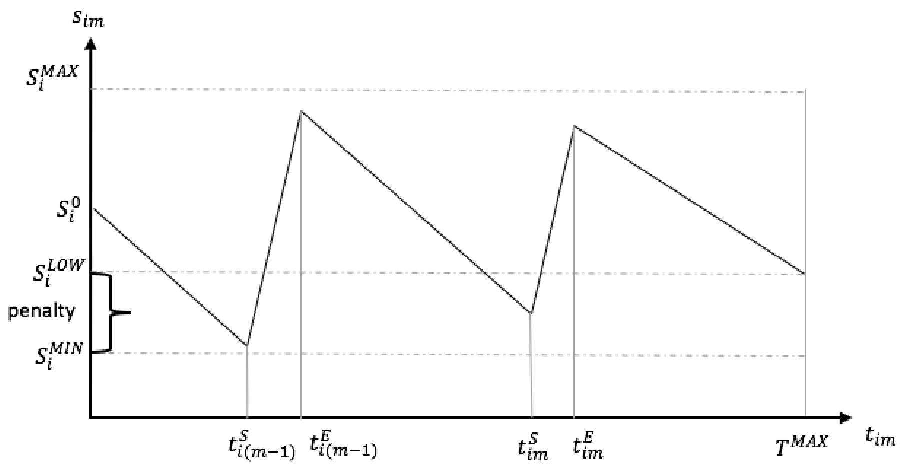

Figure 3.4: Example of inventory level during the planning horizon for an unloading port

Chapter 4

Problem Description

The problem description describes the simplified problem for seaborne transportation of fresh salmon from Norway to Europe. This chapter is based on Chapter 2, and it is necessary in order for the reader to understand the mathematical model described in Chapter 6. All the assumptions and simplifications made in order to make the mathematical model will be explained in this chapter. The main problem in this thesis is illustrated in Figure 4.1. Fresh salmon is transported with ships from different warehouses at loading ports in Norway to different warehouses in unloading ports at the Continent, which is illustrated in the figure.

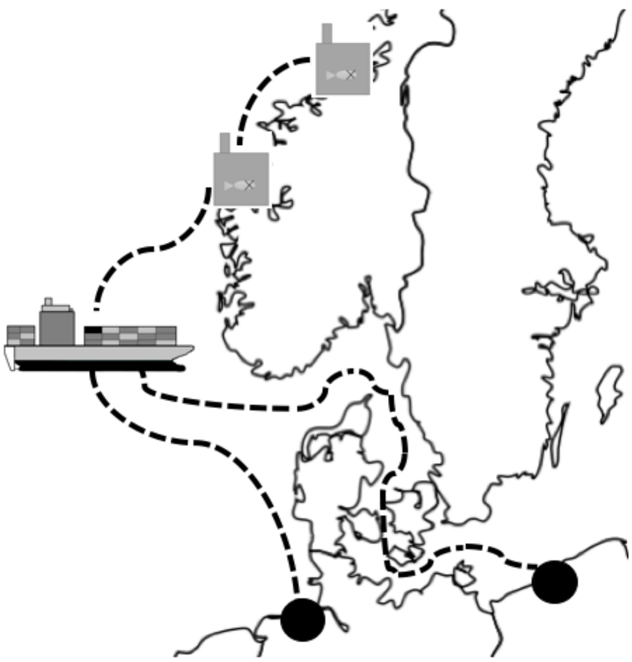

Figure 4.1: Illustrative drawing of a transportation network

Short sea shipping does not usually bring the goods all the way from the producer to the supplier so it usually has to cooperate with other land-based modes. This is called a intermodal logistics chain. This thesis has a main focus on seaborne transportation and further distribution from the unloading port is not be included. It is assumed that semi-trailers are available at the unloading ports in order to transport the salmon to the right supplier. Return load from Europe is not considered in this thesis.

Fresh fish has a fixed shelf life from being slaughtered until being consumed. Since this thesis only focus on transportation from slaughter facilities to different unloading ports, the salmon is not at its end supplier. The time from the salmon first was slaughtered until its at the unloading port is therefore not fixed, and soft time windows can be applied. Figure 4.2 illustrates this. The longer transportation time, the shorter remaining durability of the salmon for transportation further. An extra cost will therefore be added when delivery happens after a certain time. The salmon has to be delivered before it reaches the maximum number of hours after being slaughtered. After this time it is not possible for further delivery from the unloading ports and the salmon becomes waste. The model does not allow this to happen, and the delivery needs to happen before this time. The term lead time is used from the salmon first was slaughtered, until delivered at the unloading port.

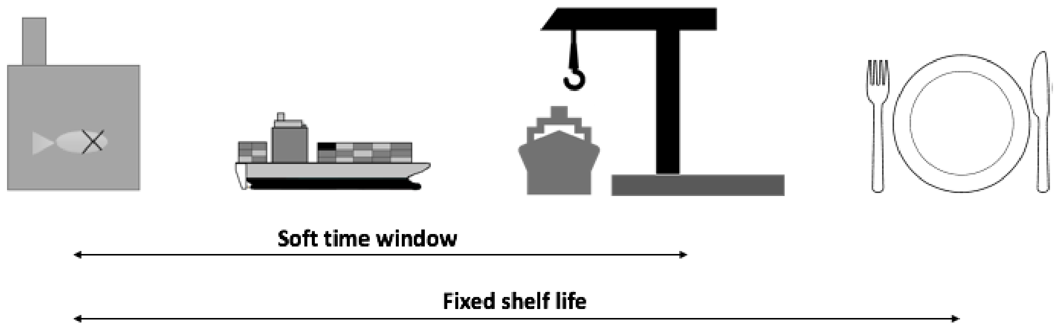

Figure 4.2: Illustration on lead time

Different slaughter facilities deliver today salmon to different markets. The salmon might go directly to stores or for further processing. For simplifications it is assumed that it does not matter where the salmon goes, and the model therefore handles one product only.

The slaughter facilities are located along the Norwegian Coast. Transportation from these facilities to Europe with vessels require ports that are able to handle them. It is assumed that the chosen ports, in both Norway and in Europe, does not have any restrictions. Since seaborne transportation for fresh salmon from Norway is not developed yet, the logistics between the slaughter facilities and potential loading ports are not developed yet either. But for the case of this problem it is assumed that the transportation time between the slaughter facilities and the loading port are neglected. Several slaughter facilities are connected to one loading port. Production rates at the slaughter facilities, are assumed according to the future. It is assumed that effective facilities are obtained in the future, and Salmars factory Innovamar is be used as reference. They can work three shifts per day, and the slaughter capacity can today be up to 900 tonnes per day (Farming company, 2016a). The production rate at the slaughter facilities is therefore assumed to be constant per hour. The inventory level in a loading port has a defined maximum level. This level will not become higher than the predefined maximum time before delivery times the production rate minus the sailing time between a loading and unloading port. Since the transportation time between the slaughter facilities and loading ports are neglected, the production rates are therefore in the loading port. Loading and unloading times are dependent on how much salmon that is loaded or unloaded, and are assumed to be equal a constant amount of tonnes salmon per hour. Short sea shipping transport goods over smaller distances and use a longer time in port. This thesis only models the model with sailing times and loading and unloading times. Other time variables regarding port times are not considered. Waiting in port is though allowed, to make the model more robust regarding transportation delay.

When the problem has several loading ports, it should not be allowable for the vessel to sail from a loading port to another loading port further north with load on board. The ship will then sail further away from the destination port and sail around with degradable load on board, which is not desirable.

The salmon starts to degrade right after it is slaughtered, and assuming constant production rates, the inventory will build up. When a ship visit a port and start to load, the oldest salmon in inventory will already be some hours old. The time the salmon has been laying in storage is the inventory level at the point of the visit divided on the production rate in that port. The oldest salmon on the vessel will always be tracked and the time from being slaughtered until being delivered cannot exceed the sat maximum time. Applying of superchilling technology can extend the shelf life of the salmon, which makes it possible for a longer time in stock before being transported.

The demand at the delivery ports is equal to the sum of the production rates at the loading ports. The demand for salmon is also assumed to be a constant amount of tonnes per hour. To avoid inventory levels higher than the demand, the inventory level needs to be kept below a maximum level. To account for uncertainty in production or unforeseen events with the vessel, the inventory level can be below zero. This coincide with the fact that the stores need to buy salmon from another buyer, and this will come in as a cost. The amount of product below the safety level is penalized by a cost for each tonnes below the level. The negative inventory level needs to be covered later.

As seen in Chapter 2 the export volumes changes throughout the year. More salmon is exported the last quarter, than the rest of the year. The model will not directly take seasonal variations into account, but it can indirectly be taken into account by running the model for different time horizons, and change the production rate in different horizons. Routing and scheduling of ships by maintaining the inventory levels is a tactical planning problem, and since several visits in the different ports are desirable, the time horizon is sat to two to three weeks. The time is broken down to hours.

Since seaborne transportation of fresh salmon is not yet developed and to account for the future, the mathematical model should be able to handle vessels of different sizes. It is therefore possible to use a fleet of heterogeneous ships in the model. When modeling inventory routing, it is hard to find the right parameters that make the model feasible. To make the model more robust the model can handle an unconstrained fleet of vessels. The ships start in an artificial node, and if a ship is not used it will stay there. At the end of the time horizon the ships will end in an artificial dummy node, which is the same node as the starting node. The ship is empty when leaving and arriving the dummy node. It is possible to load and unload in several ports. And a ship can visit several ports during the planning horizon. It is assumed that the ports are always open, and opening hours does not need to be considered.

Uncertainty in weather conditions is not considered, but if the sailing time increase it might negatively affect the freshness of the salmon, and potential waste of the product. The model will always try to minimize the cost, and by penalizing late delivery the model strive to deliver before being penalized. This can act as a safety time for uncertainty in weather conditions. It is assumed that the vessels are available throughout the period, and unavailable ships due to maintenance will not be included.

Deliver fresh salmon is the most important aspect of the model, and by penalizing delivery after a certain time, inventory cost will not directly be considered, but there will be a extra cost for the difference between delivered amount and produced amount. The costs that will be considered in this thesis are therefore sailing cost, fixed cost for use of vessels, penalty for older salmon, penalty for how much that is delivered compared to the produced amount and a penalty for falling below the safety stock level in unloading ports. In real life problem several other cost elements must be considered. The amount of products transported is often a decisive factor for the transportation cost. The model also assumes that the ships can wait in port. This cost is not included in the model for simplicity, but this cost should be included in real life.

For the remaining of this thesis the loading ports are defined as the ports that slaughter the salmon for export, and this is defined as the production rate. For the unloading ports, the consumption rates, which is also considered as demand rates are defined. A summary of the problem described in this chapter is given below. This will be the base for the mathematical model in Chapter 6.

- **Objective**
  - Minimize transportation cost
  - Fixed cost of using a ship
  - Ensure in time-deliveries and minimizing late deliveries
  - Maximizing amount of product delivered
  - Minimize low stock levels

- **Routing and scheduling**
  - Unconstrained fleet of heterogeneous ships
  - Constant speed of sailing
  - Waiting in port is allowed
  - Loading/unloading rates based on the quantity loaded/unloaded
  - Continuous time
- **Loading ports**
  - Can be several
  - Maximum stock level
  - The salmon begins its shelf life after it is slaughtered/produced
  - One product
  - Constant production rate throughout the planning period
- **Unloading ports**
  - Can be several
  - Inventory level can be negative, but is backlogged and penalized
  - The demand has a constant rate throughout the planning period

Chapter 5

Literature Review

This chapter presents relevant literature within the field of optimization that are considered relevant for this master thesis. Short sea shipping usually cooperates with other land-based modes, but since this thesis only looks at transportation from slaughter facilities to ports at the Continent, intermodal logistics chain is not addressed in this literature review. The problem considered in this thesis is a Maritime Inventory Routing Problem (MIRP), hence the characteristics of these problems are addressed. Salmon is a perishable product and literature around this is also addressed.

The literature review starts with a small chapter considering routing and scheduling of ships. Chapter 5.2 introduces relevant articles on inventory routing for both maritime transportation and land-based transportation. While, Chapter 5.3 addresses routing of perishable goods. This review is not meant to give the reader a comprehensive overview of related literature, but to provide an overview of articles that are considered the most relevant for this thesis.

5.1 Routing and Scheduling

Maritime transportation planning problems can be classified into strategic, tactical and operational problems according to the planning period. This thesis considers ship scheduling and routing and inventory ship routing, which can be classified as tactical problems (Christiansen et al., 2007). Ronen (1983) describes routing as the assignment of sequences of ports to be visited by the ships, while the term scheduling is defined as the assignment of time to the different events on a ship route. Routing and scheduling of ships is a field well studied. Ronen (1983), Ronen (1993), Christiansen et al. (2004) and Christiansen et al. (2013) are some published surveys that have reviewed ship scheduling and routing the last decades.

Maritime scheduling problems including pick-up and delivery of cargo are considered as Multi-ship Pickup and Delivery Problem (m-PDP). Fagerholt (2001) considers (m-PDP) with soft time windows. According to Fagerholt (2001) there exists no papers on ship scheduling with soft time windows in maritime literature. Scheduling with soft windows may give better schedules and significant reductions in the transportation costs, allowing time window violations for some customers. Soft time windows also reflect situations found in practice better than hard time windows.

5.2 Inventory Routing Problems

IRP can be described as the combination of vehicle routing and inventory management problems. These problems dates 30 years back (Coelho et al., 2014). According to Coelho et al. (2014) the first studies published on IRP were mainly variations on Vehicle Routing Problem (VRP) and heuristics developed to consider inventory costs. The general VRP consists of designing optimal routes for delivery from a central depot to a set of customers, subject to various constraints, such as vehicle capacity, route length, time windows and precedence relations between customers (Laporte, 2007). Christiansen et al. (2013) define a MIRP as a planning problem where an actor has the responsibility for both the inventory management at one or both ends of the maritime transportation legs, and for the routing and scheduling of the ships. The goal is to minimize the transportation cost without interrupting the production or consumption at the storages.

Andersson et al. (2010) and Coelho et al. (2014) point out that it is difficult to decide on one standard version of the IRP, because for every real application of the problem a new version is created. Andersson et al. (2010) have classified the problem into seven criteria, such as; time, demand, topology, routing, inventory, fleet composition and fleet size. While Coelho et al. (2014) also classifies the problem into the availability of information on the demand, as deterministic, stochastic or dynamic. They also include the inventory policy employed. This thesis will combine the classification scheme used by Andersson et al. (2010) and Coelho et al. (2014) to present characteristics relevant to the problem. The classification criterion's are provided in Table 5.1. Further descriptions are provided below the table.

Table 5.1: Characteristic variants of the IRP

Characteristic	**Alternatives**			
Time	Instant	Finite	Infinite	
Demand	Deterministic	Stochastic		
Topology	One-to-one	One-to-many	Many-to-many	
Routing	Direct	Multiple	Continuous	
Inventory policy	Maximum level (ML)	Order-up-to level (OU)		
Inventory	Fixed	Stock-out	Lost sale	Back-order
Fleet composition	Homogeneous	Heterogeneous		
Fleet size	Single	Multiple	Unconstrained	

Time

The planning period used can be classified into different modes. A short planning horizon is called *instant*, and it is so short that only one visit per customer is needed. The main decisions are to balance the inventory and routing cost with the costs linked to stock-outs at the customers. *Finite* time-period is considered if more than one visit at a customer may be necessary. Fixed horizon is used if there is a natural and finite end to the horizon. If there is an interaction between the time before and after the horizon, it is common to use a rolling horizon and solve the problem for a longer period than needed for the immediate decisions. The time-period is said to be *infinite* if the problem deal with distribution strategies instead of schedule decisions. Time can also be treated as discrete or continuously time periods. Discrete time-periods can handle time varying production while a continuous formulation might be more appropriate when the demand rates are fixed through the planning horizon.

Demand

If the demand parameters are known the model is considered as deterministic. While if the model incorporates uncertainty with respect to demand, it is called stochastic. The production and or consumption rates may be either a constant rate through the planning period or a time-varying demand.

Topology

The topology *one-to-one* refers to direct route between a producer and a customer. If a single facility serves several customers using a fleet of vehicle the topology is *one-to-many*. This is the dominant one for road-based vehicles, and the facility is often the depot where the vehicles start and end their routes and where the goods are stored. *Many-to-many* are more usual in a maritime setting. Usually there is no central facility and no fixed starting and ending point. The ship can load and unload at any port.

Routing

The routing component can be described as three cases. The term *direct* is used if the vehicle picks up goods at a depot and then deliver all the goods to a single customer. If a vehicle can visit more than one customer on a trip, the term *multiple* is used. Where there is no start and end and you have a pick-up and delivery setting, the term used is *continuous*.

Inventory

Inventory decisions can be handled at both suppliers and customers, or only at one end. Inventory policies define pre-established rules to refill customers inventory. *Maximum-level* (ML) policy is considered when the replenishment level is flexible. However, the maximum replenish is bounded by the capacity available at each customer. *Order-up-to level* (OU) policy is when, whenever a customer is visited, the quantity delivered fill the inventory up. How inventory management is modeled is determined by the inventory decisions. The inventory is called *fixed* if the inventory level is not allowed to fall below zero or a level based on the safety stock. Failure to satisfy the demand, can be seen as *stock-out*. A stock-out is usually followed by an emergency delivery or considered as *lost sale*. If the demand is postponed to be supplied later, it is called *back-order*.

Fleet composition

If the vehicles have the same characteristics, the fleet is *homogeneous*, if not the fleet is *heterogeneous*. A fleet consisting of only one vehicle, the term *single* is used. If the fleet consists of several vehicles and this is a constraining factor, the term *multiple* is used. If it is possible to pay for extra distribution capacity, the situation is called *unconstrained*.

Table 5.2 provides a summary of articles read and mentioned in this chapter. The papers are selected to give an overview of relevant examples that describe the typical characteristics of the IRP. A more detailed description of the articles is given below the table.

Table 5.2: Summary of relevant articles on IRP

Paper	Time	Demand	Topology	Routing	Inventory	Fleet
This paper	Cont.	Det.	M-to-m	Cont.	Back-order	Het. unconstrained
Ronen (2002)	Discr.	Det.	O-to-m	Direct	Fixed	Het.
Bertazzi et al. (2011)	Discr.	Stoch.	O-to-m	Multiple	Stock-out/ Lost sale	Hom.
Bertazzi et al. (2002)	Discr.	Det.	O-to-m	Multiple	Fixed	Hom. unconstrained
Al-Khayyal and Hwang (2007)	Cont.	Det.	M-to-m	Multiple	Fixed	Het. multiple
Agra et al. (2017)	Cont./ discr.	Det.	M-to-m	Multiple	Fixed	Het. multiple

Ronen (2002) presents a multi-product shipments-planning problem faced by producers with large volume of bulk products. The problem is to determine how much of each product and when to ship from which origin to which destination, and by which vessel. Due to uncertainties in demand and in production, prescribed safety stocks of each product have to be maintained. The problem minimizes the total shipping cost, while the safety stock and storage volume limitations are not violated. Violations of safety stock levels, will lead to a penalty cost. He approaches the problem by separating the solution of the problem into two stages. First determination of the shipments to be shipped, and second, scheduling the vessels to ship them.

Bertazzi et al. (2002) present a distribution problem with deterministic order-up-to level policies. A set of products are shipped from a supplier to several retailers, where each retailer has a minimum and maximum level of the inventory of each product. The problem is to determine which retailers to visit and the route of the vehicle for each discrete time instant. They study several objective functions corresponding to different decision policies, in order to study the impact of the objective function on the problem solution. A heuristic algorithm is used to solve the problem. The results show how relevant the goal is on the obtained solution.

Bertazzi et al. (2011) study a stochastic inventory routing problem with stock-out. The problem is to minimize the expected cost, given by the sum of the expected total inventory cost, penalty cost at the retailers and expected routing cost, to determine the best shipping strategy. A maximum inventory level is defined at the retailers and a stochastic demand has to be satisfied over a given time horizon. At each retailer, an order-up-to level policy is applied. If the inventory level becomes negative the excess demand is not backlogged, but a penalty cost will apply. They provide a dynamic programming formulation and propose a hybrid rollout algorithm.

Al-Khayyal and Hwang (2007) formulate a model for inventory constrained maritime routing and scheduling for multi-commodity liquid bulk. The problem is to decide how much of each product that should be carried between different supply-ports and demand ports for each ship. The inventory level of each product in each port must be maintained between certain levels that are set by the production and consumption rates and the storage capacities.

Agra et al. (2017) present two formulations for a short sea IRP, discrete time and continuous time. A discrete time formulation is used when the consumption rate varies. Inventory management considerations are only taken into account at the demand side. They discuss different extended formulations and valid inequalities, to reduce the linear gap. They use a commercial software to conduct a computational study to compare the various models.

5.3 Routing Problems with Perishable Products

Articles on inventory routing with perishable considerations for maritime transportation have not been found. The overview of articles with inventory routing and perishable assets are therefore with vehicles, and one for liner shipping. Table 5.3

shows a summary of the reviewed articles for perishable items. Compared to Table 5.2 some characteristics have been included. These are perishable considerations, waste and shelf life. Perishable consideration regards the main characteristics of the article regarding the product. A model can either allow the product to go bad or not. If it goes bad a cost of waste applies. The shelf life of the product is fixed if it has no time window, and soft if it has one.

Table 5.3: Summary of relevant articles that consider perishable items

Paper	Time/ Demand	Topology/ Fleet	Perishable considerations/ Waste	Inventory	Shelf life
This paper	Cont./ Det.	M-to-m / Het.	Quality time windows/ No	Back-order	Soft
Dulebenets and Ozguven (2017)	Cont./ Det.	M-to-m / Hom.	Exponential asset decay, penalty/ No	N/A	Soft
Soysal et al. (2015)	Disrc./ Stoch.	O-to-m / Het.	Cost of waste/ Yes	Back-order	Fixed
Le et al. (2013)	Discr./ Det.	O-to-m/ Hom.	Inventory cost/ No	Fixed	Fixed
Jia et al. (2014)	Discr./ Det.	O-to-m / Hom.	Quality time windows / No	Fixed	Soft TS Hard ST
Coelho and Laporte (2014)	Discr./ Det.	O-to-m/ Het.	Varying age/ No	Fixed	Fixed

According to Dulebenets and Ozguven (2017) perishability of assets has not been explicitly modeled in liner shipping by use of vessel. They present a vessel scheduling problem in liner shipping route with perishable assets, and proposes a novel mixed integer non-linear mathematical model for this problem. The model minimizes the total route service cost, including the asset decay cost, total late arrival penalty and total inventory cost. Results from the study show that the developed model will allow liner shipping companies to design efficient vessel schedules and in the meantime reduce decay of perishable assets on board the vessel.

Soysal et al. (2015) present an IRP model to account for perishability, explicit fuel consumption and demand uncertainty. The product has a fixed shelf life and if this is exceeded the product becomes waste, and a cost of waste occurs. The customers demand has to be fulfilled with a probability and demand that is not fulfilled in one period will be backlogged into the next period. The result from the model suggests that with these integration's the model can achieve significant savings in total cost.

Le et al. (2013) study the IRP for a perishable product with fixed shelf-life. The study restricts the total amount of time that products can be stored in facilities. The products will be discarded at the end of their shelf-life. The model proposes an upper bound inventory level, which is determined by the perishability constraints. They assume that vehicles travel at most one route in any time period and that customers have at most one delivery per time period. Split deliveries are therefore not allowed. They propose a column generation-based heuristic algorithm to solve the IRP for perishable goods. They believe that a branch-and-cut-and-price algorithm could work effectively for problems of small or medium size.

Jia et al. (2014) present an IRP with quality time windows and loading under discrete time. They consider time windows that are in a transit stage as well as a sales stage, to control the product quality. Soft time windows are used during the transportation stage (TS) and hard time windows during the sales stage (SS). The model determines the supplier's production plan, the retailer's delivery time and vehicle routing in each period. The results show that the vehicles loading cost and return time interval have impact on the decision variables.

Coelho and Laporte (2014b) present an age tracking approach on the inventory routing problem of perishable product with a fixed shelf life. They model the model to handle the cases where retailers always sell older items first, and where they sell fresher items first. The author's say that to their knowledge it is the first time an IRP is modeled and solved exactly under general assumptions in the context of perishable product management. The models do not require any assumption on the shape of the product revenue and inventory cost functions. They model with three echelon supply chain. Suppliers deliver products to retailers who then sell products to the end-customers. The demand of each customer in each period is the sum of product quantities of different ages. Any product whose age is higher than a number is spoiled, and it no longer appears in the inventory nor it can satisfy the demand. The results show that the profit changes drastically depending on the shape of the revenue of the product.

Chapter 6

Model Formulation

This chapter presents the mathematical model built for the problem described in Chapter 4. The model's base is created according to the formulation done by Christiansen et al. (2007) in *Chapter 4. Maritime Transportation* section *4.3.1 Inventory routing for a single product,* and it is modeled as a Mixed Integer Problem (MIP). The time is considered as continuous. Where the salmon is produced is considered as the loading port, and where the salmon is delivered in Europe is considered as the unloading port.

In the mathematical model each port, either a production port or consumption port is represented by an index i and the set of ports is given by N. Number of available ships to be routed and scheduled is given by V, indexed by v. It is assumed that all the ships can visit all the ports. The initial position for the ships is represented by *o(v)*, and the term *d(v)* represents the artificial destination port for ship v. *o(v)* and *d(v)* are modeled as the same node, and the ships are forced to travel here after their last visit. Every port can be visited several times during the planning period and M_i represents possible visits at port i. The visit number is represented by an index m, and the last possible visit at port i is $|M_i|$. The visit number m for the start and end node is made for number of ships. The set of nodes in the flow network represents the set of port visits, and each port visit is specified by *(i,m)*, $i \in N$, $m \in M_i$. The set A contains the set of visits *(i,m)* where $i \in N$ and $m \in M_i$. A_v is the set of feasible arcs for ship v including the starting node *o(v)* and dummy node $d(v)$, which is a subset of $\{i \in N, m \in M_i\} \times \{i \in N, m \in M_i\}$. Port visits and port calls are used as the same term throughout the thesis. The parameters are mostly given in tonnes and hours, if nothing else is stated. The remaining of this chapter is structured as follows: The model definitions can be seen in Chapter 6.1. The mathematical formulation with explanations to each constraint and the objective function can be seen in Chapter 6.2. Lastly, the compressed mathematical model can be seen in Chapter 6.3.

6.1 Definitions

Sets

N	Set of ports, excluding $o(v)$ and $d(v)$
M_i	Set of visit numbers for port i
V	Set of ships
A	Set of visits (i,m) where $i \in N$ and $m \in M_i$
A_v	Set of feasible visits for ship v, $A_v = A \cup \{o(v), d(v)\}$

Indices

i, j	Ports
$o(v)$	Start node for ship v
$d(v)$	Dummy end node for ship v
m, n	Visit numbers
v	Ships

Parameters

C_{ij}	Sailing cost from port i to port j
P_v	Fixed cost for using ship i
$\|M_i\|$	Maximum number of visits in port i
Q_v	Capacity for ship v
T_{ij}	Sailing time from port i to port j
T_i^Q	Loading time at port i in tonnes per hour
T^{MAX}	Length of planning period

S_i^0	Initial inventory level at port i
S_i^{MAX}	Maximum inventory level at port i
S_i^{MIN}	Minimum inventory level at port i
S_i^{LOW}	Lower safety stock at port i
I_i	Type of port i, 1 for loading ports, -1 for unloading ports and 0 for depot
R_i	Production and consumption rate for port i in tonnes per hour. Positive value for production and negative for consumption
D_1^{MAX}	The longest time since the oldest salmon was slaughtered to being delivered without being penalized
D_2^{MAX}	The maximum allowable time since the oldest salmon was slaughtered to being delivered
P_i^{TIME}	Penalty cost for delivery between D_1^{MAX} and D_2^{MAX}
P_i^{DEL}	Penalty cost for ratio between produced and delivered amount
P_i^{LOW}	Penalty cost for each tonnes of lower safety stock shortfall

Decision Variables

x_{imjnv}	1 if ship v sails from visit (i,m) to visit (j,n), 0 otherwise
w_{im}	1 if visit (i,m) is not made by any ship, 0 otherwise
l_{imv}	Total load on board ship v after service is completed for visit (i,m)
q_{imv}	Quantity loaded or unloaded by ship v during visit (i,m)
s_{im}^S	Amount of salmon in stock at the start of visit (i,m)
s_{im}^E	Amount of salmon in stock at the end of visit (i,m)
s_{im}^{LOW}	Lower safety stock shortfall at port visit (i,m)
t_{im}^S	Time at which service begins for visit (i,m)
t_{im}^E	Time at which service ends for visit (i,m)
t_{im}^{SL}	Lead time for the delivered salmon after visit (i,m)
y_{im}	Penalizing help variable

6.2 Mathematical Formulation

6.2.1 Objective Function

The objective in this model is to minimize the cost. The model is constructed to deliver fresh salmon, which is the most important aspect of the problem. The objective function contains several terms, which can be seen below:

$$\min f = \sum_{v \in V} \sum_{(i,m,j,n) \in A_v} C_{ij} x_{imjnv} \tag{6.1a}$$

$$+ \sum_{(j,n) \in A} P_v x_{o(v)jnv} \tag{6.1b}$$

$$+ \sum_{i \in N} P_i^{DEL} \left(1 - \frac{\sum_{m \in M_i} \sum_{v \in V} q_{imv}}{R_i T^{MAX} + S_i^0} \right) \tag{6.1c}$$

$$+ \sum_{(i,m) \in A} P_i^{TIME} y_{im} \tag{6.1d}$$

$$+ \sum_{(i,m) \in A} P_i^{LOW} s_{im}^{LOW} \tag{6.1e}$$

The cost of sailing from i to j is provided as C_{ij}. P_v is the fixed cost of using ship v in the planning period. P_i^{DEL} represents the penalty cost in port i for not deliver all the load produced, and it is zero for unloading ports. The parameter P_i^{TIME} represents the deteriorating cost per time when deliver salmon in the penalty window, and P_i^{LOW} represents the penalty cost for each tonnes of lower safety stock shortfall in port i. The first part (6.1a) represents the sailing cost when sailing from *(i,m)* to *(j,n)*. The second part (6.1b) is the fixed cost of using a ship. The third term (6.1c) represents the extra cost of not deliver all the salmon produced. This term is included to encourage the model to deliver as much as possible even though the inventories are satisfied. (6.1d) is the penalty for deliver salmon later than a given time, and the last term (6.1e) is the penalty of lower safety stock shortfall.

6.2.2 Routing Constraints

The following binary variables are constructed for the routing aspect of the model. x_{imjnv} is 1 if ship v sails from visit *(i,m)* to visit *(j,n)*, and 0 otherwise. Since number of visits are not predefined, the binary variable w_{im} is included. It equals 1 if no ship v visit port call *(i,m)* and 0 otherwise. This variable serves as a slack variable. The routing constraints are as follows:

$$\sum_{(j,n)\in A_v} \sum_{v\in V} x_{imjnv} + w_{im} = 1 \qquad \forall (i,m) \in A \tag{6.2}$$

$$\sum_{(j,n)\in A_v | I_j \neq -1} x_{o(v)jnv} = 1 \qquad \forall v \in V \tag{6.3}$$

$$\sum_{(j,n)\in A_v} x_{jnimv} - \sum_{(j,n)\in A_v} x_{imjnv} = 0 \qquad \forall (i,m) \in A, v \in V \tag{6.4}$$

$$\sum_{(i,m)\in A_v} x_{imd(v)v} = 1 \qquad \forall v \in V \tag{6.5}$$

$$w_{im} - w_{i(m-1)} \geq 0 \qquad \forall (i,m) \in A | m > 1 \tag{6.6}$$

Constraints (6.2) ensure that each port call is visited at most once, either by a ship or by the slack variable. Constraints (6.3)-(6.5) describe the flow on the sailing route by ship v. Constraints (6.3) ensure that every ship leaves its initial position. This constraint also makes it possible for the vessel to stay in the initial position, meaning that the ship is not used. Constraints (6.4) ensure that all subsequent visits to the different ports have equal ingoing and outgoing flow, while constraints (6.5) ensure that each ship ends in its designated end node. Since one or several calls in a port can be made by a dummy ship w_{im}, the relation in constraints (6.6) ensure that higher visiting numbers are not used unless the preceding number is also used. Thus, the model will only use the smallest subsequent numbering.

6.2.3 Loading and Unloading Constraints

The salmon is considered as a single product. The variable q_{imv} represents the quantity loaded or unloaded at port visit *(i,m)* done by ship v. The variable l_{imv}

represents the total load on board ship v just after service is completed at visit *(i,m)*. The ship can load and unload at several ports during a route, but the load on board the ship cannot exceed its capacity Q_v. The parameter I_i is equal to 1 for loading port, -1 for unloading port and 0 for start and end node.

$$x_{imjnv}(l_{imv} + I_j q_{jnv} - l_{jnv}) = 0 \qquad \forall (i,m,j,n) \in A, v \in V \tag{6.7}$$

$$q_{imv} \leq l_{imv} \leq \sum_{(j,n)\in A_v} Q_v x_{imjnv} \qquad \forall (i,m) \in A, v \in V | I_i = 1 \tag{6.8}$$

$$l_{imv} \leq \sum_{(j,n)\in A_v} Q_v x_{imjnv} - q_{imv} \qquad \forall (i,m) \in A, v \in V | I_i = -1 \tag{6.9}$$

The relation between the binary flow variable and the ship load at each port call is given in constraints (6.7). If vessel v sails from *(i,m)* to *(j,n)*, $x_{imjnv} = 1$ and the load on board when leaving *(i,m)* added or subtracted the quantity loaded or unloaded in *(j,n)* must equal the load on board vessel v when leaving *(j,n)*. This constraint is nonlinear and is not suited for direct implementation in Xpress. The constraint is linearized into two other constraints, presented in Chapter 7.1.1. Constraints (6.8) apply to loading ports and limit the quantity loaded to be less than or equal to the load on board the ship and the ship's capacity. Constraints (6.9) apply to unloading ports and limits the outgoing load from the port to be less than or equal to the ships capacity minus the quantity unloaded.

$$l_{imv} x_{imjnv} = 0 \qquad \forall (i,m,j,n) \in A, v \in V | I_i = -1, I_j = 1 \tag{6.10}$$

$$l_{imv} x_{imd(v)v} = 0 \qquad \forall (i,m) \in A, v \in V \tag{6.11}$$

To make sure that the ship does not bring load back to a loading port, constraint (6.10) is added. This constraint ensures that the ship unload all the load before leaving an unloading port and sailing to a loading port. Constraints (6.11) make sure that the vessel cannot arrive the depot with load on board the vessel. If the ship sails from visit *(i,m)* to *(j,n)*, $x_{imjnv} = 1$, the loading variable must be zero. (6.10) and (6.11) are not linear and are linearized in Chapter 7.1.1.

6.2.4 Time Constraints

To keep track of the inventory levels and ensure that the ships return to the dummy node within the end of the time horizon, time constraints are necessary. The loading or unloading times are in each port i given as T_i^Q, in tonnes per hour. Sailing time between port i and j, is provided by T_{ij}. The length of the planning period is given as T^{MAX}. The ships must complete their routes during this period, so that they can be ready for a voyage in the consecutive period.

The time variables t_{im}^S and t_{im}^E represent the starting and ending time for service in *(i,m)*, respectively. It is not necessary to have time variables for both starting and ending time, but they are both included for the readability of the results. The production and consumption rate in tonnes per hour, R_i, is positive if the ship service a loading port and negative if the ship service an unloading port. These rates are constant per hour. The formulated time constraints are given below:

$$x_{imjnv}(t_{im}^E + T_{ij} - t_{jn}^S) \leq 0 \qquad \forall (i,m,j,n) \in A, v \in V \tag{6.12}$$

$$t_{im}^S + \sum_{v \in V} \frac{q_{imv}}{T_i^Q} = t_{im}^E \qquad \forall (i,m) \in A, v \in V \tag{6.13}$$

$$t_{im}^E \leq T^{MAX} \qquad \forall (i,m) \in A \tag{6.14}$$

$$w_{im}(t_{im}^S - t_{i(m-1)}^E) \leq 0 \qquad \forall (i,m) \in A | m > 1 \tag{6.15}$$

$$t_{im}^S - t_{i(m-1)}^E \geq 0 \qquad \forall (i,m) \in A | m > 1 \tag{6.16}$$

Constraints (6.12) ensure consistency in timing of visits. If a ship sails directly between two ports, the starting time of the next visit cannot be earlier than the ending time of the previous visit plus the sailing time between the two ports. Waiting on arrival is allowed when modeling the constraint with inequality. This constraint is nonlinear, and is linearized in Chapter 7.1.1. Start and end time of a visit is related in constraints (6.13). It ensures that a visit ends when loading or unloading is finished. The ending time of every visit must be less or equal to the planning horizon, as seen in constraints (6.14). If a visit is not made by a ship, the starting time should be sat equal to the ending time of the preceding visit number. Constraints (6.15) ensure this. This constraint is linearized in Chapter 7.1.1. To prevent service overlap in port the starting time of a visit must be greater or equal to the end time of the previous visit. Constraints (6.16) take care of this.

6.2.5 Perishable Constraints

To make sure that the salmon delivered is fresh, some constraints on the perishability are made. The variable t^{SL}_{im} represents the time since the salmon first was slaughtered and began its shelf life. As described in the problem description in Chapter 4 the time from slaughtering to delivery in unloading ports is modeled with soft time windows, and the parameters D^{MAX}_1 and D^{MAX}_2 describe the penalty window. D^{MAX}_2 describes the maximum allowable time since the salmon was slaughtered until it has to be delivered, while D^{MAX}_1 describes the time since the salmon was slaughtered until its delivered without being penalized. The following constraints are included for perishable considerations:

$$t^{SL}_{im} \geq \frac{s^S_{im}}{R_i} + \frac{q_{imv}}{T^Q_i} \qquad \forall (i,m) \in A, v \in V | I_i = 1 \tag{6.17}$$

$$t^{SL}_{jn} \geq \Big(\frac{s^S_{jn}}{R_j} + \frac{q_{jnv}}{T^Q_j}\Big) x_{imjnv} \qquad \forall (i,m,j,n) \in A, v \in V | I_i = 1, I_j = 1 \tag{6.18}$$

$$t^{SL}_{jn} \geq (t^{SL}_{im} + t^E_{jn} - t^E_{im}) x_{imjnv} \qquad \forall (i,m,j,n) \in A, v \in V | I_i = 1, I_j = 1 \tag{6.19}$$

$$x_{imjnv}(t^{SL}_{im} + t^E_{jn} - t^E_{im} - t^{SL}_{jn}) = 0 \qquad \forall (i,m,j,n) \in A, v \in V | I_j = -1 \tag{6.20}$$

$$t^{SL}_{im} \leq D^{MAX}_2 \qquad \forall (i,m) \in A \tag{6.21}$$

$$y_{im} \geq t^{SL}_{im} - D^{MAX}_1 \qquad \forall (i,m) \in A | I_j = -1 \tag{6.22}$$

Constraints (6.17) track the time the salmon was first slaughtered in a loading port. The inventory level before service begin in *(i,m)*, divided on the production rate plus the time it takes to load the ship, will give the time since the oldest salmon was produced after the ship is loaded. When the ship sails between two loading ports it is important to track the oldest product. This is ensured with constraints (6.18) and (6.19), by choosing the largest number of either the time in stock in *(i,m)* plus the time it takes to arrive *(j,n)* and load here, or the time in stock in *(j,n)*. The largest number is chosen for further sailing and constraints (6.20) keep track of the time when sailing to an unloading port. Constraints (6.18)-(6.20) are all nonlinear and are linearized in Chapter 7.1.1. Constraints (6.21) set the maximum time the salmon is allowed to use from it first was slaughtered to being unloaded at an unloading port. To penalize delivery time over the given value D_1^{MIN} in the objective function, constraint (6.22) is made. y_{im} is not allowed to be less than zero, and is therefore only given a value when t_{im}^{SL} is larger or equal to D_1^{MIN}.

6.2.6 Inventory Constraints

To ensure that the stock of salmon does not exceed a certain level an upper stock level is defined, S_i^{MAX}. In unloading ports, the demand can be negative, which indicates that the customers need to buy the salmon elsewhere. An extra cost will therefore apply for negative inventory in unloading ports, and the shortage in demand must be satisfied later. A lower safety stock level is therefore defined as S_i^{LOW}, while the minimum value of the inventories are defined as S_i^{MIN}. The variable s_{im}^{LOW} represents the lower safety stock shortfall of the product at the beginning of service for visit *(i,m)*. Initial stock level is denoted by S_i^0. The stock level is represented by s_{im}^S before a visit *(i,m)* is made, and the stock level after the visit is s_{im}^E. It would have been sufficient with only one variable, as for the time variables. The inventory constraints are as follows:

$$S_i^0 + R_i t_{im}^S = s_{im}^S \qquad \forall (i,m) \in A | m = 1 \qquad (6.23)$$

$$s_{i(m-1)}^E + R_i (t_{im}^S - t_{i(m-1)}^E) = s_{im}^S \qquad \forall (i,m) \in A | m > 1 \qquad (6.24)$$

$$s_{im}^S + R_i (t_{im}^E - t_{im}^S) - I_i \sum_{v \in v} q_{imv} = s_{im}^E \qquad \forall (i,m) \in A \qquad (6.25)$$

Constraints (6.23) set the stock level at the start of the first visit, as the initial stock level plus or minus the production or consumption, respectively. Constraints (6.24) relate the stock at the end of a visit to the stock at the start of the next visit, by considering either the production or consumption that takes place between the visits. Constraints (6.25) ensure that the stock at the end of service equals the stock level at start of service, modified for production or consumption and the quantity loaded or unloaded in the time period.

$$s_{im}^{E} + R_i(T^{MAX} - t_{im}^{E}) \leq S_i^{MAX} \qquad \forall (i,m) \in A | m = |M_i|, I_i = 1 \tag{6.26}$$

$$s_{im}^{E} + R_i(T^{MAX} - t_{im}^{E}) \geq S_i^{LOW} \qquad \forall (i,m) \in A | m = |M_i|, I_i = -1 \tag{6.27}$$

$$q_{imv} \leq \sum_{(j,n) \in A_v} S_i^{MAX} x_{imjnv} \qquad \forall (i,m) \in A, v \in V | I_i = -1 \tag{6.28}$$

$$s_{im}^{S} + s_{im}^{LOW} \geq S_i^{LOW} \qquad \forall (i,m) \in A | I_i = -1 \tag{6.29}$$

To make sure that the stock levels does not exceed or fall below the inventory level throughout the whole planning period, constraints (6.26) and (6.27) are added for loading and unloading ports, respectively. Constraints (6.28) restrict the load unloaded at port i to be less or equal to the maximum inventory level for port i. This constraint does not consider how much load that is already in stock, but another constraint takes care of the inventories. Constraints (6.29) calculate the safety stock shortfall in unloading ports.

$$s_{im}^{E} \geq S_i^{MIN} \qquad \forall (i,m) \in A | I_i = 1 \tag{6.30}$$

$$s_{im}^{S} \leq S_i^{MAX} \qquad \forall (i,m) \in A | I_i = 1 \tag{6.31}$$

$$s_{im}^{S} \geq S_i^{MIN} \qquad \forall (i,m) \in A | I_i = -1 \tag{6.32}$$

$$s_{im}^{E} \leq S_i^{MAX} \qquad \forall (i,m) \in A | I_i = -1 \tag{6.33}$$

Constraints (6.30) and (6.31) ensure that the stock variables are within the minimum and maximum inventory levels for the loading ports, while constraints (6.32) and (6.32) ensure that the stock variables are within the minimum and maximum inventory levels for the unloading ports.

6.2.7 Variable Constraints

The variable constraints can be seen in (6.34)-(6.42). (6.34) and (6.35) are the binary requirements on respectively the flow variable and the slack variable. (6.36)-(6.41) are all continuous variables and are defined for values larger or equal to zero. To reduce the number of variables made, (6.41) is only made for unloading ports. The inventory level in port i before service starts, s^S_{im}, is defined for negative values and the variable is free, as seen in (6.42).

$$x_{imjnv} \in \{0,1\} \qquad (i,m) \in A_v, (j,n) \in A_v, v \in V \tag{6.34}$$

$$w_{im} \in \{0,1\} \qquad (i,m) \in A \tag{6.35}$$

$$q_{imv}, l_{imv} \geq 0 \qquad (i,m) \in A, v \in V \tag{6.36}$$

$$t^S_{im}, t^E_{im} \geq 0 \qquad (i,m) \in A \tag{6.37}$$

$$t^{SL}_{im} \geq 0 \qquad (i,m) \in A \tag{6.38}$$

$$s^E_{im} \geq 0 \qquad (i,m) \in A \tag{6.39}$$

$$s^{LOW}_{im} \geq 0 \qquad (i,m) \in A \tag{6.40}$$

$$y_{im} \geq 0 \qquad (i,m) \in A | I_i = -1 \tag{6.41}$$

$$s^S_{im} \quad \text{free} \qquad (i,m) \in A \tag{6.42}$$

6.3 Compressed Mathematical Model

The compressed mathematical model can be seen below. The variable constraints are not included here, but can be seen in Chapter 6.2.7.

Sets

N	Set of ports, excluding $o(v)$ and $d(v)$
M_i	Set of visit numbers for port i
V	Set of ships
A	Set of visits (i,m) where $i \in N$ and $m \in M_i$
A_v	Set of feasible visits for ship v, $A_v = A \cup \{o(v), d(v)\}$

Indices

i, j	Ports
$o(v)$	Start node for ship v
$d(v)$	Dummy end node for ship v
m, n	Visit numbers
v	Ships

Parameters

C_{ij}	Sailing cost from port i to port j
P_v	Fixed cost for using ship i
$\|M_i\|$	Maximum number of visits in port i
Q_v	Capacity for ship v
T_{ij}	Sailing time from port i to port j
T_i^Q	Loading time at port i in tonnes per hour
T^{MAX}	Length of planning period
S_i^0	Initial inventory level at port i
S_i^{MAX}	Maximum inventory level at port i
S_i^{MIN}	Minimum inventory level at port i
S_i^{LOW}	Lower safety stock at port i

I_i	Type of port i, 1 for loading ports, -1 for unloading ports and 0 for depot
R_i	Production and consumption rate for port i per unit time. Positive value for production and negative for consumption
D_1^{MAX}	The longest time since the oldest salmon was slaughtered to being delivered without being penalized
D_2^{MAX}	The maximum allowable time since the oldest salmon was slaughtered to being delivered
P_i^{TIME}	Penalty cost for delivery between D_1^{MAX} and D_2^{MAX}
P_i^{DEL}	Penalty cost for ratio between produced and delivered amount
P_i^{LOW}	Penalty cost for each tonnes of lower safety stock shortfall

Decision Variables

x_{imjnv}	1 if ship v sails from visit (i,m) to visit (j,n), 0 otherwise
w_{im}	1 if visit (i,m) is not made by any ship, 0 otherwise
l_{imv}	Total load on board ship v after service is completed for visit (i,m)
q_{imv}	Quantity loaded or unloaded by ship v during visit (i,m)
s_{im}^S	Amount of salmon in stock at the start of visit (i,m)
s_{im}^E	Amount of salmon in stock at the end of visit (i,m)
s_{im}^{LOW}	Lower safety stock shortfall at port visit (i,m)
t_{im}^S	Time at which service begins for visit (i,m)
t_{im}^E	Time at which service ends for visit (i,m)
t_{im}^{SL}	Lead time for the delivered salmon after visit (i,m)
y_{im}	Penalizing help variable

Model Formulation

$$\min f = \sum_{v \in V} \sum_{(i,m,j,n) \in A_v} C_{ij} x_{imjnv} + \sum_{(j,n) \in A} P_v x_{o(v)jnv} \tag{6.43}$$

$$+ \sum_{i \in N} P_i^{DEL} \left(1 - \frac{\sum_{m \in M_i} \sum_{v \in V} q_{imv}}{R_i T^{MAX} + S_i^0} \right) + \sum_{(i,m) \in A} P_i^{TIME} y_{im} \tag{6.44}$$

$$+ \sum_{(i,m) \in A} P_i^{LOW} s_{im}^{LOW} \tag{6.45}$$

$$\sum_{(j,n) \in A_v} \sum_{v \in V} x_{imjnv} + w_{im} = 1 \qquad \forall (i,m) \in A \tag{6.46}$$

$$\sum_{(j,n) \in A_v | I_j \neq -1} x_{o(v)jnv} = 1 \qquad \forall v \in V \tag{6.47}$$

$$\sum_{(j,n) \in A_v} x_{jnimv} - \sum_{(j,n) \in A_v} x_{imjnv} = 0 \qquad \forall (i,m) \in A, v \in V \tag{6.48}$$

$$\sum_{(i,m) \in A_v} x_{imd(v)v} = 1 \qquad \forall v \in V \tag{6.49}$$

$$w_{im} - w_{i(m-1)} \geq 0 \qquad \forall (i,m) \in A | m > 1 \tag{6.50}$$

$$x_{imjnv}(l_{imv} + I_j q_{jnv} - l_{jnv}) = 0 \qquad \forall (i,m,j,n) \in A, v \in V \tag{6.51}$$

$$q_{imv} \leq l_{imv} \leq \sum_{(j,n) \in A_v} Q_v x_{imjnv} \qquad \forall (i,m) \in A, v \in V | I_i = 1 \tag{6.52}$$

$$l_{imv} \leq \sum_{(j,n) \in A_v} Q_v x_{imjnv} - q_{imv} \qquad \forall (i,m) \in A, v \in V | I_i = -1 \tag{6.53}$$

$$l_{imv}x_{imjnv} = 0 \qquad \forall(i,m,j,n) \in A, v \in V | I_i = -1, I_j = 1 \tag{6.54}$$

$$l_{imv}x_{imd(v)v} = 0 \qquad \forall(i,m) \in A, v \in V \tag{6.55}$$

$$x_{imjnv}(t^E_{im} + T_{ij} - t^S_{jn}) \leq 0 \qquad \forall(i,m,j,n) \in A, v \in V \tag{6.56}$$

$$t^S_{im} + \sum_{v \in V} \frac{q_{imv}}{T^Q_i} = t^E_{im} \qquad \forall(i,m) \in A, v \in V \tag{6.57}$$

$$t^E_{im} \leq T^{MAX} \qquad \forall(i,m) \in A \tag{6.58}$$

$$w_{im}(t^S_{im} - t^E_{i(m-1)}) \leq 0 \qquad \forall(i,m) \in A | m > 1 \tag{6.59}$$

$$t^S_{im} - t^E_{i(m-1)} \geq 0 \qquad \forall(i,m) \in A | m > 1 \tag{6.60}$$

$$t^{SL}_{im} \geq \frac{s^S_{im}}{R_i} + \frac{q_{imv}}{T^Q_i} \qquad \forall(i,m) \in A, v \in V | I_i = 1 \tag{6.61}$$

$$t^{SL}_{jn} \geq \Big(\frac{s^S_{jn}}{R_j} + \frac{q_{jnv}}{T^Q_j}\Big)x_{imjnv} \qquad \forall(i,m,j,n) \in A, v \in V | I_i = 1, I_j = 1 \tag{6.62}$$

$$t^{SL}_{jn} \geq (t^{SL}_{im} + t^E_{jn} - t^E_{im})x_{imjnv} \qquad \forall(i,m,j,n) \in A, v \in V | I_i = 1, I_j = 1 \tag{6.63}$$

$$x_{imjnv}(t^{SL}_{im} + t^E_{jn} - t^E_{im} - t^{SL}_{jn}) = 0 \qquad \forall(i,m,j,n) \in A, v \in V | I_j = -1 \tag{6.64}$$

$$t^{SL}_{im} \leq D^{MAX}_2 \qquad \forall(i,m) \in A \tag{6.65}$$

$$y_{im} \geq t^{SL}_{im} - D^{MAX}_1 \qquad \forall(i,m) \in A | I_j = -1 \tag{6.66}$$

$$S_i^0 + R_i t_{im}^S = s_{im}^S \qquad \forall (i,m) \in A | m = 1 \tag{6.67}$$

$$s_{i(m-1)}^E + R_i(t_{im}^S - t_{i(m-1)}^E) = s_{im}^S \qquad \forall (i,m) \in A | m > 1 \tag{6.68}$$

$$s_{im}^S + R_i(t_{im}^E - t_{im}^S) - I_i \sum_{v \in v} q_{imv} = s_{im}^E \qquad \forall (i,m) \in A \tag{6.69}$$

$$s_{im}^E + R_i(T^{MAX} - t_{im}^E) \leq S_i^{MAX} \qquad \forall (i,m) \in A | m = |M_i|, I_i = 1 \tag{6.70}$$

$$s_{im}^E + R_i(T^{MAX} - t_{im}^E) \geq S_i^{LOW} \qquad \forall (i,m) \in A | m = |M_i|, I_i = -1 \tag{6.71}$$

$$q_{imv} \leq \sum_{(j,n) \in A_v} S_i^{MAX} x_{imjnv} \qquad \forall (i,m) \in A, v \in V | I_i = -1 \tag{6.72}$$

$$s_{im}^S + s_{im}^{LOW} \geq S_i^{LOW} \qquad \forall (i,m) \in A | I_i = -1 \tag{6.73}$$

$$s_{im}^E \geq S_i^{MIN} \qquad \forall (i,m) \in A | I_i = 1 \tag{6.74}$$

$$s_{im}^S \leq S_i^{MAX} \qquad \forall (i,m) \in A | I_i = 1 \tag{6.75}$$

$$s_{im}^S \geq S_i^{MIN} \qquad \forall (i,m) \in A | I_i = -1 \tag{6.76}$$

$$s_{im}^E \leq S_i^{MAX} \qquad \forall (i,m) \in A | I_i = -1 \tag{6.77}$$

Chapter 7

Computational Study

The mathematical model described in Chapter 6 is written in Mosel Version 4.0.3 and implemented in Xpress-IVE Version 1.24.12, 64 Bit. A Intel Xeon 3.33GHz computer with a 32 GB RAM is used to solve the optimization.

The remainder of this chapter is structured as follows. To be able to implement the mathematical model into Xpress IVE several adjustments are necessary and these are described in Chapter 7.1. Test cases with the necessary input data are provided in Chapter 7.2 and the results from the test cases can be seen in Chapter 7.3. The source code for the mathematical model can be seen in Appendix B.

7.1 Model Adjustments

7.1.1 Linearizing Constraints

In order to solve the model with Xpress-Optimizer, non linear constraints have to be linearized. This is done by using the Big M method (Williams, 2013). The complete and compressed mathematical model with the performed linearizations can be found in Appendix A.

$$l_{imv} + I_j q_{jnv} - l_{jnv} - Q_v(1 - x_{imjnv}) \leq 0 \qquad \forall (i,m,j,n) \in A, v \in V \tag{7.1a}$$

$$l_{imv} + I_j q_{jnv} - l_{jnv} + Q_v(1 - x_{imjnv}) \geq 0 \qquad \forall (i,m,j,n) \in A, v \in V \tag{7.1b}$$

Constraints (7.1a) and (7.1b) linearize constraints (6.7). Since Q_v is the largest value that $l_{imv} + I_j q_{jnv} - l_{jnv}$ can take, and $-Q_v$ is the smallest, these constraints are only binding if ship v sails from node *(i,m)* to *(j,n)*.

$$l_{imv} \leq Q_v(1 - x_{imjnv}) \qquad \forall (i,m,j,n) \in A, v \in V | I_i = -1, I_j = 1 \tag{7.2}$$

$$l_{imv} \leq Q_v(1 - x_{imd(v)v}) \qquad \forall (i,m) \in A, v \in V \tag{7.3}$$

Constraints (7.2) and (7.3) are the linearized constraints of respectively (6.10) and (6.11). If ship v sails from *(i,m)* to *(j,n)*, $x_{imjnv} = 1$, and l_{imv} must be less or equal to zero. Since l_{imv} cannot be negative these constraints make sure that the load on board the ship equals zero if you go from an unloading port to a loading port, or to the depot.

$$t^E_{im} + T_{ij} x_{imjnv} - t^S_{jn} - T^{MAX}(1 - x_{imjnv}) \leq 0 \qquad \forall (i,m,j,n) \in A, v \in V \tag{7.4}$$

Constraint (6.12) is linerized to constraints (7.4). Since (6.12) is an inequality, only one relation is needed. The largest value of $t^E_{im} - t^S_{jn}$ is T^{MAX}, and the constraint is only binding if a ship v sails from node *(i,m)* to *(j,n)*.

$$t^S_{im} - t^E_{i(m-1)} - T^{MAX}(1 - w_{im}) \leq 0 \qquad \forall (i,m) \in A | m > 1 \tag{7.5}$$

Constraints (7.5) linearize constraints (6.15). It ensures equality between t^S_{im} and $t^E_{i(m-1)}$ if a visit is not made.

$$t_{jn}^{SL} + T^{MAX} \geq \frac{s_{jn}^{S}}{R_j} + \frac{q_{jnv}}{T_j^{Q}} + T^{MAX} x_{imjnv} \quad \forall (i,m,j,n) \in A, v \in V | I_i = 1, I_j = 1 \tag{7.6}$$

$$t_{jn}^{SL} + T^{MAX} \geq t_{im}^{SL} + t_{jn}^{E} - t_{im}^{E} + T^{MAX} x_{imjnv} \quad \forall (i,m,j,n) \in A, v \in V | I_i = 1, I_j = 1 \tag{7.7}$$

Constraints (7.6) linearize constraints (6.18), and constraints (7.7) linearize constraints (6.19). These constraints are only binding if ship v sails from *(i,m)* to *(j,n)* and both ports are loading ports.

$$t_{im}^{SL} + t_{jn}^{E} - t_{im}^{S} - t_{jn}^{SL} - T^{MAX}(1 - x_{imjnv}) \leq 0 \qquad \forall (i,m,j,n) \in A, v \in V | I_j = -1 \tag{7.8a}$$

$$t_{im}^{SL} + t_{jn}^{E} - t_{im}^{S} - t_{jn}^{SL} + T^{MAX}(1 - x_{imjnv}) \geq 0 \qquad \forall (i,m,j,n) \in A, v \in V | I_j = -1 \tag{7.8b}$$

Constraint (6.20) is linearized with constraints (7.8a) and (7.8b). These are only binding if a ship sail to an unloading port.

7.1.2 Variable Reduction

Increasing the problem size, increases the solution time, and in order to solve the problem within reasonable time, elimination of variables are essential. In this thesis the variable x_{imjnv} is reduced by only creating it for $i \neq j$, and for $i, j = 0$ which is the start and end node. The variable y_{im} is also reduced by only being created for unloading ports. The variables related to the stock level and time, are both created for start and end time of loading or unloading in the different ports. They are derived from other variables and parameters with equality constraints, and this will increase the model size. A more detailed discussion around the variables can be seen in Chapter 8.

7.2 Test Case

In order to investigate whether the model is suitable as a decision support tool and get insight into seaborne transportation, a case study is created for this thesis. The input data presented in this chapter is to the best possible extent based on real data. Some assumptions have also been made, and the reasoning around these are explained. The input parameters to the model are converted to a text file that serves as input to Xpress.

The initial thought on how the case study was going to be executed is presented below. However, due to problems running the solver, not all of the cases were achieved tested. Discussion regarding this is further presented in Chapter 8.

7.2.1 Ports

With the purpose of showing what the model is capable to and to get insight into seaborne transportation of fresh salmon, some ports are chosen for the case study. Three ports are chosen in Norway and two ports in Europe. Different combinations of these are tested together. The Norwegian ports are Rørvik, Hitra and Bergen, and the ports in Europe are, Cuxhaven in Germany and Zeebrugge in Belgium. Table 7.1 shows the distance between the different ports.

Table 7.1: Distance between the ports, given in nm (Dataloy, 2016)

	Rørvik	**Hitra**	**Bergen**	**Zeebrugge**	**Cuxhaven**
Rørvik (1)	-	-	-	-	-
Hitra (2)	98	-	-	-	-
Bergen (3)	360	279	-	-	-
Zeebrugge (4)	911	829	574	-	-
Cuxhaven (5)	767	685	430	299	-

The different ports with numbers from table 7.1, are illustrated with their locations in Figure 7.1.

Figure 7.1: Illustration of the different port locations

Different scenarios are modelled in this thesis. The different scenarios can be seen in 7.2. All the scenarios are to be tested with increasing production rates. The thought behind scenario 2 and 3 is to see the difference when two loading ports further north are used compared to two loading ports further south. Scenario 4 is created to see what happens when two unloading ports are used, compared to two loading ports, when the rates are the same.

Table 7.2: Different scenarios for the case study

Scenario	Loading port(s)	Unloading port(s)
1	Hitra	Cuxhaven
2	Rørvik, Hitra	Cuxhaven
3	Hitra, Bergen	Cuxhaven
4	Hitra	Cuxhaven, Zebrugge

7.2.2 Production and Demand

Since seaborne transportation of fresh salmon from Norway to Europe does not exist yet, some assumptions regarding the production and demand rates are made. The mathematical model is modelled with inventory levels in both loading and unloading ports. The sum of the production rates in loading ports must therefore be equal to the sum of the consumption rates in unloading ports. 62.8% of the produced amount of salmon was in 2015 exported to Europe (SSB, 2016). It is assumed that the chosen loading ports in Norway handle the produced amount of salmon in the respective county, and that 62.8% of this number is exported to Europe. Since there are eight slaughter facilities spread around in Hordaland county, it is assumed that Bergen handle half of the produced amount. These assumptions give the rates for export in Table 7.3.

Table 7.3: Production rates for export in different ports

Port	[tonnes/year]	[tonnes/hour]
Rørvik	66430	7.6
Hitra	89507	10
Bergen	51319	5.8

It will take 18 days to produce the amount of salmon needed to fill a ship with capacity of 2500 tonnes, when using the production rate for export from Bergen. This is infeasible, since the salmon's shelf life is 3 weeks. The rates must therefore increase for seaborne transport to be possible. The production is expected to increase, and the model will be tested with larger number than the ones in Table 7.3.

7.2.3 Inventory Levels

The initial inventory level chosen for each port is important for the feasibility of the solution. The initial level in an unloading port must be chosen so the levels do not fall below the minimum level, before a vessel has the possibility to arrive. The initial inventory level in a loading port must be chosen such that the first vessel can deliver salmon to an unloading port so it does not exceed the maximum lead time. For the maximum lead time to be valid, the maximum inventory level in loading ports must be chosen so it does not make a stricter constraint than the lead time. The inventory levels are dependent on the production and consumption rates, sailing time and maximum lead time given. All these values must have the right combination in order to give a feasible solution. Some of the chosen inventory levels can be seen in Appendix C and D for two and three ports, respectively.

7.2.4 Time

The model is tested for a planning period from two to three weeks. Production and consumption rates are assumed constant within this period. The model defines time in hours, and all the parameters are therefore given in hours.

7.2.5 Loading and Unloading Rates

The loading and unloading rates are assumed to be similar in all ports. As mentioned in Chapter 2, a container ship is utilized. It is assumed that the container crane has a capacity of 25 containers every hour, and one container takes around 19 tonnes of salmon. This gives a loading or unloading rate of 475 tonnes salmon per hour.

7.2.6 Shelf Life

The salmon's shelf life from being slaughtered until being consumed is about three weeks according to Marine Harvest (2016). The maximum lead time for salmon is 5 days when transported with semi-trailers from Norway to Europe (Farming company, 2016a). Transportation by semi-trailers transport the salmon from door to door. Accepting a shorter durability in the stores by utilizing ships, a maximum lead time when the salmon is transported with ships is therefore assumed to be 6 days. As it is desirable to deliver as fresh fish as possible, lead time after 4 days will be penalized. Applying super-chilling technology can increase the durability with one week. The model is not tested with this technology, but can easily be included. Table 7.4 shows the parameters for the model and their respective implemented values.

Table 7.4: Maximum time since the salmon was slaughtered until it has to be delivered, and the penalty window

Parameter	**[hours]**
D_1^{MAX}	96
D_2^{MAX}	144

7.2.7 Cost Aspect

The ship's travelling costs are obtained from transportation cost models developed by Grønland (2011). A container ship with 8500 DWT is used as reference, but the vessel used in this thesis is around 5000 DWT. The reference vessel by Grønland (2011) is therefore too large, but the cost is reduced accordingly. The traveling cost of 2000 NOK per hour is used and calculated for the different distances between the ports. For simplifications, a sailing speed of 13 knots is used in these calculations, and the costs will not be changed if another sailing speed is used. The calculated travelling costs can be seen in Table 7.5. The costs are rounded to the nearest thousand. The costs of loading and unloading are not included in this rate, and therefore not considered.

Table 7.5: Cost of traveling between the ports, given in [NOK]

	Rørvik	**Hitra**	**Bergen**	**Zeebrugge**	**Cuxhaven**
Rørvik	-	-	-	-	-
Hitra	15 000	-	-	-	-
Bergen	55 000	43 000	-	-	-
Zeebrugge	140 000	128 000	88 000	-	-
Cuxhaven	118 000	105 000	66 000	46 000	-

Working with IRP, it is hard to find the right parameters to make the model run, and not become infeasible. To make the model more robust it can handle an unconstrained fleet of vessels. A fixed cost for each ship that is used will therefore be added. The cost is based on Pedersen et al. (2006). The fleet of vessels modelled in this case study are container vessels around 5000 DWT, and based on a ship design from Pedersen et al. (2006). This vessel has a length of 136 m over all, and can carry 2700 tonnes of salmon. Ships around this size will be tested. Different vessels are tested in the model and can be seen in Table 7.6.

Table 7.6: Fixed cost for different vessels

Type	**Capacity [tonnes]**	**Speed [knots]**	**Fixed cost [NOK]**
1	2500	15	900 000
2	2000	15	800 000

In order to deliver as much as possible of the produced amount, a penalty cost of 500 000 NOK is assumed for the ratio between delivered versus produced amount. Longer lead time will lead to a more degraded product. Striving to deliver a fresh product, a penalty cost of 1000 NOK per hour for delivery after a certain time is assumed. Based on Pedersen et al. (2006) the freight income is 950 NOK per tonnes. The penalty cost for not delivering before the demand fall below zero in unloading ports, is assumed to be approximately twice as much, 2000 NOK per tonnes. The costs are summarized in Table 7.7 with the same parameter formulations as in the model.

Table 7.7: Penalty costs

Parameter	**Cost**	
P_i^{DEL}	500 000	[NOK]
P_i^{TIME}	1000	[NOK/tonnes]
P_i^{LOW}	2000	[NOK/tonnes]

7.3 Results of Computational Study

The results of the computational study are presented in this chapter. First, solutions from running the model for scenario 1 are presented. Running the model for the other scenarios have shown to be difficult. Thus, not all of the scenarios are presented. A further discussion of this is presented in Chapter 8. Running the model yields an optimal sailing pattern, optimal time for deliveries, optimal inventory levels and amount of load for each delivery, which are presented in this chapter.

7.3.1 Scenario 1: Hitra-Cuxhaven

Scenario 1 is the first scenario that goes from one loading port to one delivery port, and Table 7.8 presents the solutions to different parameters investigated for transportation from Hitra to Cuxhaven. The rate represents the production and consumption rate of salmon for loading and unloading ports respectively, given in tonnes per hour. L represents the loading port and U represents the unloading port.

Period given in hours represents the length of the planning period the model is run for. Rate and period are input parameters. Ships represent number of the different types of ship with capacity and speed, that are needed in the solution. The lead time represents the longest lead time during the time horizon for the deliveries and is from the salmon first was slaughtered until delivered. Amount of salmon delivered versus produced is represented in the next column. And last the cost of the solution presented in 1000 NOK. The input file for test number 1 can be seen in Appendix C.

Table 7.8: Different solutions when sailing between Hitra and Cuxhaven

Test No.	**Rate**		**Period**	**Ships**			**Lead time**	$\frac{\textbf{Delivered}}{\textbf{Produced}}$	**Cost**
	L	U	[hours]	#	[tonnes]	[knots]	[hours]		[*10^3 NOK]
1	35	-35	500	1	2500	15	128	85 %	3372
				1	2000	15			
2	50	-50	336	2	2500	15	106.5	86 %	3857
				1	2000	15			

Table 7.8 shows the difference between test number 1 and 2. When the production rate is increased to 50 tonnes per hour, three ships are used in the best solution. The longest lead time for test number 2 is decreased compared to solution number 1, and the solution uses a maximum of four and a half days to deliver the salmon, from first slaughtered, throughout the whole planning period. The delivered versus produced amount through the period can not be compared, because the test numbers are tested with different periods. The period is reduced, with a hope of reducing the solution time. The same fixed cost of using a vessel has however been assumed equal even thought the periods are not the same. Test number 1 is solved to optimality after 9987 seconds, but not test number 2. This solution was stopped after almost 14 hours, with a gap of 10.7 %. This will be discussed further in Chapter 8.

Test No. 1

Running the model for test number 1 from Table 7.8 yields the sailing pattern and visiting sequences illustrated in Figure 7.2. $(0, 1)$ and $(0, 2)$ are the start and end nodes for vessel 1 and vessel 2 respectively. The optimal solution for this case, is when the vessels take every other visiting number each. The first number represents the port number, and the last number the visit number in that port. Port number

1 is the loading port, Hitra, and port number 2 is the unloading port, Cuxhaven. Both ports are visited eight times during the planning period of 3 weeks.

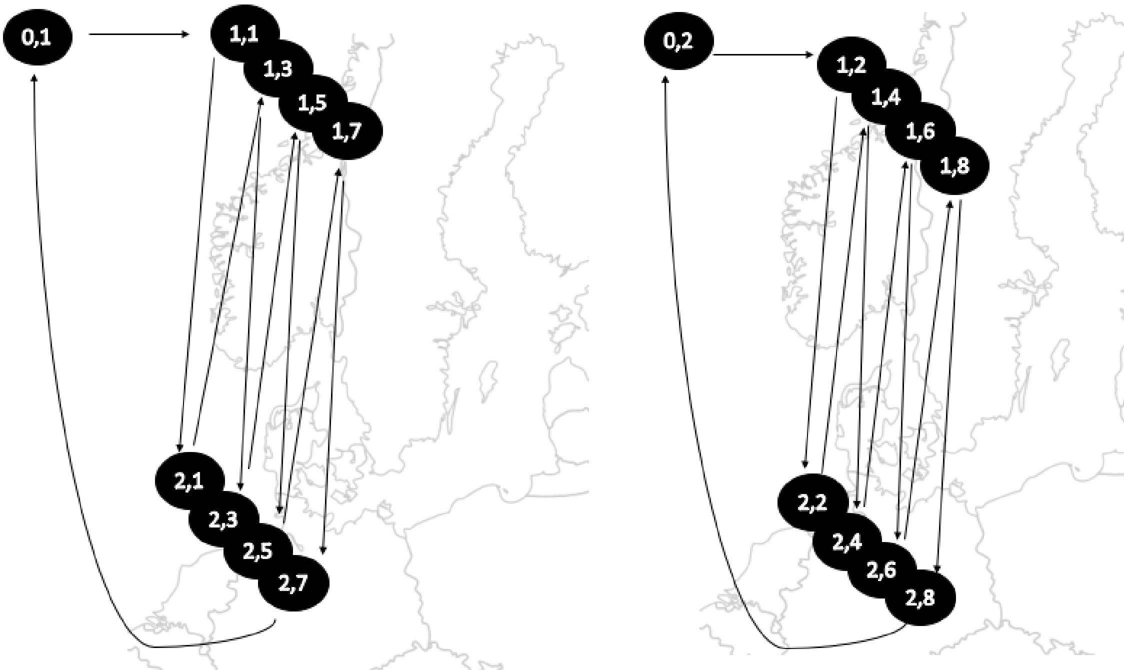

Figure 7.2: Visiting sequence between the two ports for two vessels

The inventory level in Hitra, throughout the planning period is illustrated in Figure 7.3. The upper bound is illustrated with the horizontal dotted line. The dotted line stretching from zero to the upper bound at the end of the planning period, illustrates the increase in inventory from the last visit until the end of the planning period. As seen from the figure, the inventory level at the end of the period is at its maximum value. The model finds the best solution based on the inventory level at the end of the planning period.

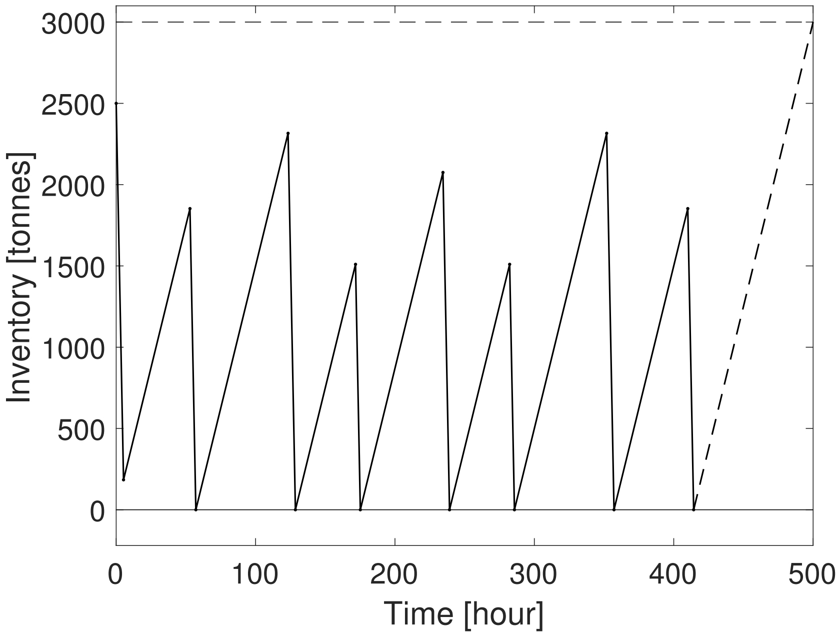

Figure 7.3: Inventory level in loading port, Hitra, throughout the planning period

The inventory level in Cuxhaven is illustrated in Figure 7.4. The dotted horizontal lines illustrate the upper and lower bounds of the inventory.

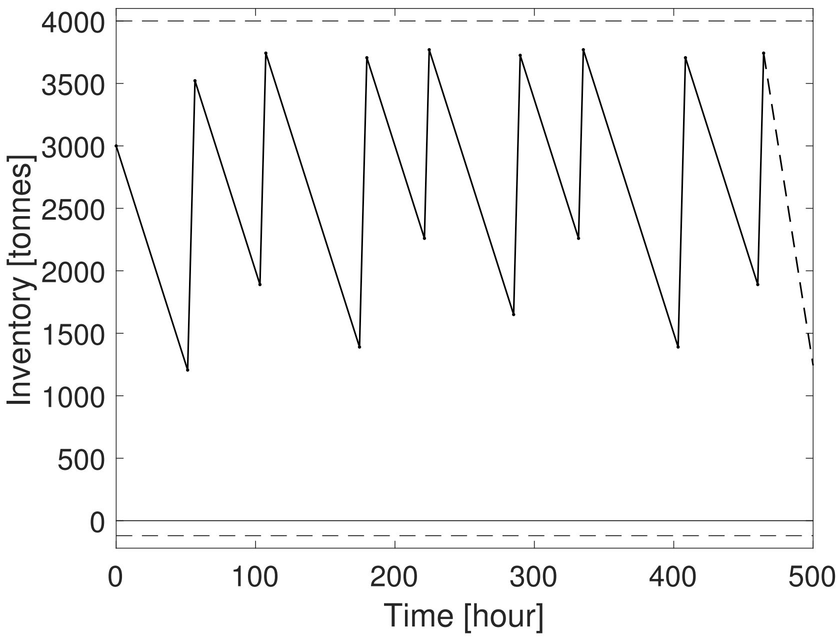

Figure 7.4: Inventory level in unloading port, Cuxhaven, throughout the planning period

The load on board vessel 1 and vessel 2 when sailing from the loading port to the unloading port, can be seen in Table 7.9. Vessel 1 has utilized 97.4% of its capacity throughout the planning period, while vessel 2 has utilized 90.75% of its capacity.

Table 7.9: Load on board the vessels when sailing from the loading port for different visit numbers

Visit Number	Vessel Number	Load on board
1	1	2500
2	2	2000
3	1	2500
4	2	1630
5	1	2240
6	2	1630
7	1	2500
8	2	2000

7.3.2 Scenario 2: Rørvik-Hitra-Cuxhaven

Scenario 2 is created with two loading ports and one unloading port, to see what happens when number of loading ports are increased. Some tested solutions for scenario 2 with Rørvik, Hitra and Cuxhaven as the ports, can be seen in Table 7.10.

Table 7.10: Different solutions for scenario 2

Test no.	Rate			Integer solutions found	Running time
	L	L	U	#	[seconds]
1	40	40	-80	0	50 000
2	40	30	-70	2	13 000

For test number 1, no solution is found after running the model for 50 000 seconds. The input parameters used for this test number, can be seen in Appendix D. The maximum lead time for test number 2 is increased with 16 hours, to see if a solution can be obtained faster. After 2967 and 3019 seconds two solutions are found. They both have an optimality gap of 76.5%, and after running the model for 13 000 seconds, the optimality gap has not become any smaller.

Due to problems running the solver, the other scenarios are not tested. This is discussed further in Chapter 8.

Chapter 8

Discussion

This chapter discusses the mathematical model and the results of the test cases presented in Chapter 7.3. The mathematical model and its application as a decision support tool will be discussed, and how it can be improved to better provide a representation of the real-life problem.

Solution time is an important factor if the model shall be used as a decision tool. Planners often need to make fast decisions, and the model should provide a near optimal or an optimal solution within a short period of time. A heuristic solution is slightly more expensive than the optimal one, but the time the model uses to solve the problem to optimality might be more expensive than the increased cost of utilizing the heuristic solution. Therefore, a near-optimal solution is considered to provide an adequate result for the planners.

It is observed that the solution time is decreased if the quantity loaded and unloaded and the load on board the vessel is modeled as integer numbers, and not non-integers. However, the solution space decreases when the problem is modeled with integers and less flexibility is provided. Modeling should therefore be with non-integers. It is also observed that the model uses less time when the cost parameters are scaled down. Using the same other parameters, and dividing the costs on 1000, the solution time goes from 287 seconds to 200 seconds.

Another observation is that the solution time takes less time when another formulation regarding the flow constraints is used. Constraints (6.3)- (6.5) are in the implemented model summed over just the different loading and unloading ports, and not the start and end node. The start and end node are added and subtracted, respectively. This can be seen in Appendix B and constraints number A.5- A.7. The solution time takes less time with the other formulation, because the model has less ports to sum over, which makes it an easier formulation.

When running the model, the solution time is extremely sensitive to changes in input parameters. Increasing one of the costs with 10% the model can use a significant longer time. Therefore, it has been hard to run the model with different parameters, when the running time of the model is not known. More work should therefore be performed on making the model formulation stricter.

The model is constructed to handle unconstrained fleet of vessel, but increasing the number of vessels as input, increases the solution time. Number of vessels have therefore tried to be chosen as few as possible when running the model, to decrease the solution time.

Several variables are derived from other variables. This applies for the inventory variables and time variables. These variables are constructed for start and end time of service in port, and are constructed with the purpose to ease the reading of the results. They are defined as equality constraints and if only one of these variables were used the number of constraints will also be reduced, as it is no longer necessary with the constraints that link the start and end time and the start and end inventory levels for each visit. Reducing the variables will require some remodeling, where these variables are used.

The most important aspect of the model formulation in this thesis is the maximum lead time. This variable controls the solution, since the salmon has to be delivered before a certain time. It also controls how the inventory level in loading ports look like. The inventory level will not become higher than the maximum lead time subtracted with the sailing time from the loading port to the unloading port. The inventory level in loading ports will always try to minimize the level, because the model aims to minimize the lead time. The inventory level in unloading ports will also be controlled by the lead time and levels in loading ports, if the unloading ports do not require load before the maximum lead time. As seen from Figure 7.4 for inventory level in the unloading port, the levels are over all high. This is based on a high initial level and that the restriction regarding the lead time is stricter than the inventory levels.

When salmon is delivered within the penalized time window, the amount of salmon unloaded is not considered. The reason for this is that it is not feasible to linearize two continuous variables, and the time variable for the penalized time window and the quantity unloaded are both continuous. Therefore, the quantity unloaded is not considered. It would be more realistic if the model considered both the amount unloaded and the time when it is penalized, since later delivery of more salmon should cost more, than later delivery with less salmon.

When modeling an inventory problem, number of visits in each port are not determined in advance and number of visits are determined based on how many trips the vessel needs to take to keep the inventories satisfied. A penalty cost for how much that is delivered in the period compared to how much that is produced is added trying to get as many visits as possible. Inventory routing modeling, will sometimes also give time between the end of the planning period and the last visit the vessel made, when it is impossible to make another trip with the vessel. It is therefore a weakness in the evaluation number regarding the percent delivered versus produced amount of salmon in the period. It might be that the vessels have delivered everything they can in the period, and are not able to make one more visit before the end of the period. The evaluation number will therefore show a smaller percent delivered, because of the produced amount from the last visit until the end of the horizon that the vessel is not able to deliver.

Several costs and penalty costs are utilized in the objective function striving to reach the best possible solution. These are tried to be based on realistic values, but changes in the costs and the difference between them, can have a significant impact on the solution. The input parameters are therefore a significant source of error.

As stated in the problem description it is not logical to sail further north to load more salmon when the ship already has load on board, but restrictions on sailing routes has not been included in the mathematical model. The most important aspect of the model is the lead time of the salmon, and as long as this time is under the maximum limit, the model finds the best sailing route to minimize the cost. A restriction on allowable routes can easily be added to the model.

The model only considers ships in the solution. A more realistic model formulation would be to include the possibility for transportation with semi-trailers as well. Then the vehicles and ships could cooperate to find the best solution for transportation of salmon, and minimize the lead time. Seasonal variations have not directly been included in the model formulation. A solution can be to use the same fleet throughout the year for the lowest export volumes and use semi-trailers when the export is higher.

The transportation between the slaughter facilities and the loading ports have been excluded in the model formulation. Producing enough salmon to fill up a vessel takes time. The outmost slaughter facilities can therefore transport the salmon to the loading port, and let the facilities nearby just produce the salmon straight to the vessel. The salmon is thus not waiting a longer time if it is produced at an outskirt facility, and the transportation will not change the salmon's lead time. Thus, the model formulation makes a realistic assumption regarding excluding the transportation between the slaughter facilities and the loading ports.

The model can be run with different maximum lead time, and the solution will change subsequently. Superchilling technology has not been tested for the model, but the maximum lead time can just be increased to alter for this technology.

The production rate of salmon is assumed to be constant per hour, according to the large slaughter facilities that can slaughter salmon in three shifts per day. Inventory routing with continuous time formulation is therefore modeled. The problem could have been modeled with a different optimization method. It might be that a pick-up and delivery method, where different salmon batches have to be delivered within a certain time window, would provide a better solution approach than inventory routing. Inventory routing with discrete time formulation could also be considered. Then the production rate and demand rate could be different in different periods, and the salmon could maximum be a certain period in stock before it had to be delivered. However, inventory routing with continuous time formulation is considered to be a good solution method when transporting a perishable product, as fresh salmon. The model tracks the salmon throughout the whole planning period, and controls the time since it first was slaughtered. This is considered as an advantage when transporting a perishable product.

Running of the model with one loading port and one unloading port and with different production rates gives different solutions. From a production of 35 tonnes per hour to 50 tonnes per hour, one more ship is required in the solution. The maximum lead time is also decreased when increasing the production rate. This means that number of vessels in the solution serve as an important factor for the lead time. As long as the production rate is high, and vessels are available, the model strive to decrease the lead time. It should be mentioned that running the model with 50 tonnes per hour, did not give an optimal solution. After running the model for 14 hours, it still had an optimality gap of 10.7%. It is hard to say if another solution could have been obtained if the model would be run for a longer time. And if it would be possible to get a solution with two vessels and the same lead time as for test number 1. The difference between the costs for using a vessel and the penalty cost for late delivery has not been investigated.

After running the model for 14 hours with three ports, no solutions are found with maximum lead time of six days. When modeling IRP there are many parameters that must comply with each other in order to get a feasible solution. Increasing the port number makes it harder to choose right inventory input values for the ports. It is therefore hard to know if the tested parameters are infeasible, or if the model needs a longer time to obtain a solution. When the model is solved with higher lead time, a solution is obtained, but with an optimality gap of 76.5%. It is hard to say if the other test number with shorter lead time was infeasible or if the model had to use longer time to be solved.

The reasons behind why no other solutions are tested and achieved, and high optimality gaps, are that IRP has a high complexity. Tailor made methods are usually developed to solve the problem, and are often based on heuristics or decomposition techniques (Agra et al., 2017). Different valid inequalities can be developed in order to get a smaller optimality gap, or solve increased sizes of problems to optimality. This is shown by Coelho and Laporte (2014a). They use test instances from Coelho and Laporte (2013) to prove that their algorithm is able to yield a similar or even lower average gaps within almost one sixth of the running time. This shows the importance of tight formulations and valid inequalities in order to obtain a solution for IRP. However, the focus in this master thesis has been on developing a mathematical model for routing of fresh salmon, and solution methods has not been in focus. A lot of time was used in order for the model developed to work for two ports. The model formulation developed is not tight enough to obtain solutions within reasonable time for three ports, and fewer results than hoped have been obtained. The aim of the thesis was to get insight into seaborne transport and look at different scenarios, but with a complex developed model this has proven to be hard. Due to time limitations, solution methods have not further been investigated. The model has therefore weaknesses regarding the use as a decision tool. More work should be performed on making the constraints and formulations stricter and tighter, in order for this model to become a decision tool for seaborne transportation of fresh salmon.

To summarize this chapter, several extensions and improvements can be made to make the model more optimal to use as a decision support tool to investigate seaborne transportation of fresh salmon. The model is made to be an exact model with main focus on the salmon's shelf life, and not the model's ability to find solutions. Stricter and tighter formulations and development of valid inequality are necessary to make the IRP developed more solvable.

Chapter 9

Conclusion

Production of salmon is increasing and the road network has overall high load. Transportation with vessels can therefore contribute to less semi-trailers on the roads. Seaborne transport is complex and subject to several variables and input parameters. Optimization might therefore serve as a useful decision support tool in order to gain insight on how seaborne transport might be performed. The problem in this thesis is modeled as an IRP, where both inventory levels and routing are considered.

The developed IRP model considers a perishable product. The model tracks the salmon from it first was slaughtered, and constraints regarding maximum allowable time from slaughter to delivery are constructed. Thus, the model gives important information regarding the lead time of salmon.

The solution from the case study yields that two vessels are required to transport salmon between Hitra and Cuxhaven, when 35 tonnes of salmon is produced every hour. The time since the salmon first was slaughtered until delivered is throughout the period maximum five and a half days. Increasing the production rate to 50 tonnes per hour, three vessels are necessary for transportation. The lead time also decreases, to a maximum of four and a half days. The computational study shows that the number of vessels necessary to keep the inventories satisfied and minimizing the lead time and cost are dependent on the production rate.

The aim of this thesis was to utilize optimization to gain insight into seaborne transport of fresh salmon from Norway to Europe. A lot of time has been used on building a model that consider a perishable product, and due to a developed model with high complexity fewer solutions than planned for has been obtained. IRP has a high complexity and tighter formulations are necessary in order to achieve solutions. However, the model developed presents a new type of model that considers maritime inventory routing of a perishable product. Further work on making tighter formulations and valid inequalities are therefore recommended in order to get the model to become a decision support tool in the future.

Bibliography

Agra, A., Christiansen, M., and Delgado, A. (2017). Discrete time and continuous time formulations for a short sea inventory routing problem. *International Multidisciplinary Journal to Promote Optimization Theory Applications in Engineering Sciences*, 18(1):269–297.

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(1):106–130.

Andersson, H., 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(9):1515–1536.

Bertazzi, L., Bosco, A., Guerriero, F., and Laganà, D. (2011). A stochastic inventory routing problem with stock-out. *Transportation Research Part C*.

Bertazzi, L., Paletta, G., and Speranza, M. (2002). Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science*, 36(1):119.

Chima, C. M. (2011). Supply-chain management issues in the oil and gas industry. *Journal of Business Economics Research (JBER)*, 5(6).

Christiansen, M., Fagerholt, K., Nygreen, B., and Ronen, D. (2007). *Chapter 4 Maritime Transportation*, volume Volume 14, pages 189–284. Elsevier.

Christiansen, M., Fagerholt, K., Nygreen, B., and Ronen, D. (2013). Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, 228(3):467–483.

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

Coelho, L., Cordeau, J.-F., and Laporte, G. (2014). Thirty years of inventory routing. *Transportation Science*, 48(1):1–19.

Coelho, L. C. and Laporte, G. (2013). The exact solution of several classes of inventory-routing problems. *Computers Operations Research*, 40(2):558–565.

Coelho, L. C. and Laporte, G. (2014a). Improved solutions for inventory-routing problems through valid inequalities and input ordering. *International Journal of Production Economics*, 155:391–397.

Coelho, L. C. and Laporte, G. (2014b). Optimal joint replenishment, delivery and inventory management policies for perishable products. *Computers and Operations Research*, 47:42–52.

Dataloy (2016). Distance table. Web Page. Accessible at: `http://dataloy.com/`, Accessed Sept. 5. 2016.

Dulebenets, M. A. and Ozguven, E. E. (2017). Vessel scheduling in liner shipping: Modeling transport of perishable assets. *International Journal of Production Economics*, 184:141–156.

ECSA (2016). Short sea shipping: The full potential yet to be unleashed. Report.

Fagerholt, K. (2001). Ship scheduling with soft time windows: An optimisation based approach. *European Journal of Operational Research*, 131(3):559–571.

Farming company (2016a). Capacities of processing facilities. Email Correspondence Nov. 5th 2016.

Farming company (2016b). Transportation time from hitra to different costumers at the continent. Email Correspondence Nov. 20th 2016.

Fiskeridirektoratet (2015). Nøkkeltall fra norsk havbruksnæring. *Nøkkeltall fra norsk havbruksnæring.*

Fiskeriøkonomisk, O. and Norsk, f. (2013). Norsk fiskerinæring. *Norsk fiskerinæring*, (5):77–79.

Grønland, S. E. (2011). Kostnadsmodeller for transport og logistikk.

Hanssen, T. E. S., Solvoll, G., Nerdal, S., Runderem, O., Alteren, L., and Mathisen, T. A. (2014). Transportstrømmer av fersk laks og ørret fra norge. Report

SIB-rapport;5/2014, Handelshøgskolen i Bodø, Senter for innovasjon og bedriftsøkonomi.

Holmyard, N. (2016). Offshore aquaculture becoming an economic reality. Web Page. Accessible at: http://www.seafoodsource.com/commentary/offshore-aquaculture-becoming-an-economic-reality, Accessed Nov. 10. 2016.

Høy, E., Sunde, L., and Bjelland, H. (2013). Hvor mye laks er det egentlig i merden? *Norsk Fiskeoppdrett*, 38(2).

Jia, T., Li, X., Wang, N., and Li, R. (2014). Integrated inventory routing problem with quality time windows and loading cost for deteriorating items under discrete time. *Mathematical Problems in Engineering*, 2014.

Kaale, L. D. (2014). *Modelling and ice crystallization/recrystallization of foods in superchilling technology : superchilling of Atlantic salmon (Salmo salar).* Thesis. Avhandling (ph.d.) - Norges teknisk-naturvitenskapelige universitet, Trondheim, 2014.

Laks (2016). Et blikk fremover. Web Page. Accessible at: http://laks.no/lakseeventyret/, Accessed Nov. 15. 2016.

Laporte, G. (2007). What you should know about the vehicle routing problem. *Naval Research Logistics (NRL)*, 54(8):811–819.

Le, T., Diabat, A., Richard, J.-P., and Yih, Y. (2013). A column generation-based heuristic algorithm for an inventory routing problem with perishable goods. *Optimization Letters*, 7(7):1481–1502.

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

Marine Harvest (2016). Salmon farming industry handbook. Report, Marine Harvest ASA.

Mattilsynet (2016). Hvilke temperaturkrav er det til ferske fiskerivarer? Web Page. Accessible at: http://www.mattilsynet.no/mat_og_vann/produksjon_av_mat/fisk_og_sjomat/mottak_tilvirking_fisk/hvilke_temperaturkrav_er_det_til_ferske_fiskerivarer.22022, Accessed Nov. 20. 2016.

Nahmias, S. and SpringerLink (2011). *Perishable Inventory Systems*. International Series in Operations Research Management Science, 160. Springer US : Imprint: Springer, 1. edition.

Nord-Trøndelag Havn Rørvik, I. (2016). Blå mat-grønn logistikk. [Conference 3-March 2016].

Norwegian Food Safety Authority (2016). Fishery establishments. Report.

Pedersen, R., Mørkve, O. T., Storøy, J., Heskestad, Ivar Nordtvedt, T. S., Skjong, B., Stenersen, D., Skjølsvik, K. O., Sandaas, I., and Edvarsen, T. (2006). Høykapasitet sjøverts logistikksystem for sjømat - hovedrapport. Report SFH80 F063082, SINTEF Fiskeri og havbruk AS.

Ronen, D. (1983). Cargo ships routing and scheduling: Survey of models and problems. *European Journal of Operational Research*, 12(2):119–126.

Ronen, D. (1993). Ship scheduling: The last decade. *European Journal of Operational Research*, 71(3):325–333.

Ronen, D. (2002). Marine inventory routing: Shipments planning. *The Journal of the Operational Research Society*, 53(1):108–114.

Salmar (2017). Et blikk fremover. Web Page. Accessible at: `http://www.salmar.no/en/innovamar-from-dream-to-reality`, Accessed Feb. 27. 2017.

Sinkaberghansen (2016). Videreforedling. Web Page. Accessible at: `http://www.sinkaberghansen.no/produkter/videreforedling`, Accessed Sept. 5. 2016.

Soysal, M., Bloemhof-Ruwaard, J., Haijema, R., and van Der Vorst, J. (2015). Modeling an inventory routing problem for perishable products with environmental considerations and demand uncertainty. *International Journal of Production Economics*, 164:118.

SSB (2016). Akvakultur, 2015, foreløpige tall. Web Page. Accessible at: `https://www.ssb.no/jord-skog-jakt-og-fiskeri/statistikker/fiskeoppdrett/aar-forelopige/2016-06-02`, Accessed Sept. 7. 2016.

The Norwegian Ministry of Trade, I. and Fisheries (2014). Aquaculture. Web Page. Accessible at: `http://www.fisheries.no/aquaculture/Aquaculture/#.WB4QEuHhBTY`, Accessed Sept. 9. 2016.

Waters, D. (2003). *Logistics : an introduction to supply chain management*. Palgrave Macmillan, Basingstoke.

Williams, H. P. (2013). *Model Building in Mathematical Programming*. Wiley, Hoboken, 5th ed. edition.

Appendix A

Mathematical Model

Sets

N	Set of ports, excluding $o(v)$ and $d(v)$
M_i	Set of visit numbers for port i
V	Set of ships
A	Set of visits (i,m) where $i \in N$ and $m \in M_i$
A_v	Set of feasible visits for ship v, $A_v = A \cup \{o(v), d(v)\}$

Indices

i, j	Ports
$o(v)$	Start node for ship v
$d(v)$	Dummy end node for ship v
m, n	Visit numbers
v	Ships

Parameters

C_{ij}	Sailing cost from port i to port j
P_v	Fixed cost for using ship i
$\|M_i\|$	Maximum number of visits in port i
Q_v	Capacity for ship v
T_{ij}	Sailing time from port i to port j
T_i^Q	Loading time at port i in tonnes per hour
T^{MAX}	Length of planning period
S_i^0	Initial inventory level at port i
S_i^{MAX}	Maximum inventory level at port i
S_i^{MIN}	Minimum inventory level at port i
S_i^{LOW}	Lower safety stock at port i
I_i	Type of port i, 1 for loading ports, -1 for unloading ports and 0 for depot
R_i	Production and consumption rate for port i per unit time. Positive value for production and negative for consumption
D_1^{MAX}	The longest time since the oldest salmon was slaughtered to being delivered without being penalized
D_2^{MAX}	The maximum allowable time since the oldest salmon was slaughtered to being delivered
P_i^{TIME}	Penalty cost for delivery between D_1^{MAX} and D_2^{MAX}
P_i^{DEL}	Penalty cost for ratio between produced and delivered amount
P_i^{LOW}	Penalty cost for each tonnes of lower safety stock shortfall

Decision Variables

x_{imjnv}	1 if ship v sails from visit (i,m) to visit (j,n), 0 otherwise
w_{im}	1 if visit (i,m) is not made by any ship, 0 otherwise
l_{imv}	Total load on board ship v after service is completed for visit (i,m)
q_{imv}	Quantity loaded or unloaded by ship v during visit (i,m)

s_{im}^{S} Amount of salmon in stock at the start of visit (i,m)
s_{im}^{E} Amount of salmon in stock at the end of visit (i,m)
s_{im}^{LOW} Lower safety stock shortfall at port visit (i,m)

t_{im}^{S} Time at which service begins for visit (i,m)
t_{im}^{E} Time at which service ends for visit (i,m)

t_{im}^{SL} Lead time for the delivered salmon after visit (i,m)
y_{im} Penalizing help variable

Model Formulation

$$\min f = \sum_{v \in V} \sum_{(i,m,j,n) \in A_v} C_{ij} x_{imjnv} + \sum_{(j,n) \in A} P_v x_{o(v)jnv} \tag{A.1}$$

$$+ \sum_{i \in N} P_i^{DEL} \left(1 - \frac{\sum_{m \in M_i} \sum_{v \in V} q_{imv}}{R_i T^{MAX} + S_i^0} \right) + \sum_{(i,m) \in A} P_i^{TIME} y_{im} \tag{A.2}$$

$$+ \sum_{(i,m) \in A} P_i^{LOW} s_{im}^{LOW} \tag{A.3}$$

$$\sum_{(j,n) \in A_v} \sum_{v \in V} x_{imjnv} + w_{im} = 1 \qquad \forall (i,m) \in A \tag{A.4}$$

$$\sum_{(j,n) \in A_v | I_j \neq -1} x_{o(v)jnv} = 1 \qquad \forall v \in V \tag{A.5}$$

$$\sum_{(j,n) \in A_v} x_{jnimv} - \sum_{(j,n) \in A_v} x_{imjnv} = 0 \qquad \forall (i,m) \in A, v \in V \tag{A.6}$$

$$\sum_{(i,m) \in A_v} x_{imd(v)v} = 1 \qquad \forall v \in V \tag{A.7}$$

$$w_{im} - w_{i(m-1)} \geq 0 \quad \forall (i,m) \in A | m > 1 \tag{A.8}$$

$$l_{imv} + I_j q_{jnv} - l_{jnv} - Q_v(1 - x_{imjnv}) \leq 0 \quad \forall (i,m,j,n) \in A, v \in V \tag{A.9}$$

$$l_{imv} + I_j q_{jnv} - l_{jnv} + Q_v(1 - x_{imjnv}) \geq 0 \quad \forall (i,m,j,n) \in A, v \in V \tag{A.10}$$

$$q_{imv} \leq l_{imv} \leq \sum_{(j,n) \in A_v} Q_v x_{imjnv} \quad \forall (i,m) \in A, v \in V | I_i = 1 \tag{A.11}$$

$$l_{imv} \leq \sum_{(j,n) \in A_v} Q_v x_{imjnv} - q_{imv} \quad \forall (i,m) \in A, v \in V | I_i = -1 \tag{A.12}$$

$$l_{imv} \leq Q_v(1 - x_{imjnv}) \quad \forall (i,m,j,n) \in A, v \in V | I_i = -1, I_j = 1 \tag{A.13}$$

$$l_{imv} \leq Q_v(1 - x_{imd(v)v}) \quad \forall (i,m) \in A, v \in V \tag{A.14}$$

$$t^E_{im} + T_{ij} x_{imjnv} - t^S_{jn} - T^{MAX}(1 - x_{imjnv}) \leq 0 \quad \forall (i,m,j,n) \in A, v \in V \tag{A.15}$$

$$t^S_{im} + \sum_{v \in V} \frac{q_{imv}}{T^Q_i} = t^E_{im} \quad \forall (i,m) \in A, v \in V \tag{A.16}$$

$$t^E_{im} \leq T^{MAX} \quad \forall (i,m) \in A \tag{A.17}$$

$$t^S_{im} - t^E_{i(m-1)} - T^{MAX}(1 - w_{im}) \leq 0 \quad \forall (i,m) \in A | m > 1 \tag{A.18}$$

$$t^S_{im} - t^E_{i(m-1)} \geq 0 \quad \forall (i,m) \in A | m > 1 \tag{A.19}$$

$$t^{SL}_{im} \geq \frac{s^S_{im}}{R_i} + \frac{q_{imv}}{T^Q_i} \quad \forall (i,m) \in A, v \in V | I_i = 1 \quad \text{(A.20)}$$

$$t^{SL}_{jn} + T^{MAX} \geq \frac{s^S_{jn}}{R_j} + \frac{q_{jnv}}{T^Q_j} + T^{MAX} x_{imjnv} \quad \forall (i,m,j,n) \in A, v \in V | I_i = 1, I_j = 1 \quad \text{(A.21)}$$

$$t^{SL}_{jn} + T^{MAX} \geq t^{SL}_{im} + t^E_{jn} - t^E_{im} + T^{MAX} x_{imjnv} \quad \forall (i,m,j,n) \in A, v \in V | I_i = 1, I_j = 1 \quad \text{(A.22)}$$

$$t^{SL}_{im} + t^E_{jn} - t^S_{im} - t^{SL}_{jn} - T^{MAX}(1 - x_{imjnv}) \leq 0 \quad \forall (i,m,j,n) \in A, v \in V | I_j = -1 \quad \text{(A.23)}$$

$$t^{SL}_{im} + t^E_{jn} - t^S_{im} - t^{SL}_{jn} + T^{MAX}(1 - x_{imjnv}) \geq 0 \quad \forall (i,m,j,n) \in A, v \in V | I_j = -1 \quad \text{(A.24)}$$

$$t^{SL}_{im} \leq D^{MAX}_2 \qquad \forall (i,m) \in A \qquad \text{(A.25)}$$

$$y_{im} \geq t^{SL}_{im} - D^{MAX}_1 \qquad \forall (i,m) \in A | I_j = -1 \qquad \text{(A.26)}$$

$$S^0_i + R_i t^S_{im} = s^S_{im} \qquad \forall (i,m) \in A | m = 1 \qquad \text{(A.27)}$$

$$s^E_{i(m-1)} + R_i(t^S_{im} - t^E_{i(m-1)}) = s^S_{im} \qquad \forall (i,m) \in A | m > 1 \qquad \text{(A.28)}$$

$$s^S_{im} + R_i(t^E_{im} - t^S_{im}) - I_i \sum_{v \in v} q_{imv} = s^E_{im} \qquad \forall (i,m) \in A \qquad \text{(A.29)}$$

$$s^E_{im} + R_i(T^{MAX} - t^E_{im}) \leq S^{MAX}_i \qquad \forall (i,m) \in A | m = |M_i|, I_i = 1 \tag{A.30}$$

$$s^E_{im} + R_i(T^{MAX} - t^E_{im}) \geq S^{LOW}_i \qquad \forall (i,m) \in A | m = |M_i|, I_i = -1 \tag{A.31}$$

$$q_{imv} \leq \sum_{(j,n) \in A_v} S^{MAX}_i x_{imjnv} \qquad \forall (i,m) \in A, v \in V | I_i = -1 \tag{A.32}$$

$$s^S_{im} + s^{LOW}_{im} \geq S^{LOW}_i \qquad \forall (i,m) \in A | I_i = -1 \tag{A.33}$$

$$s^E_{im} \geq S^{MIN}_i \qquad \forall (i,m) \in A | I_i = 1 \tag{A.34}$$

$$s^S_{im} \leq S^{MAX}_i \qquad \forall (i,m) \in A | I_i = 1 \tag{A.35}$$

$$s^S_{im} \geq S^{MIN}_i \qquad \forall (i,m) \in A | I_i = -1 \tag{A.36}$$

$$s^E_{im} \leq S^{MAX}_i \qquad \forall (i,m) \in A | I_i = -1 \tag{A.37}$$

$$x_{imjnv} \in \{0, 1\} \qquad (i,m) \in A_v, (j,n) \in A_v, v \in V \tag{A.38}$$

$$w_{im} \in \{0, 1\} \qquad (i,m) \in A \tag{A.39}$$

$$q_{imv}, l_{imv} \geq 0 \qquad (i,m) \in A, v \in V \tag{A.40}$$

$$t^{SL}_{im}, s^{LOW}_{im}, s^E_{im}, t^S_{im}, t^E_{im} \geq 0 \qquad (i,m) \in A \tag{A.41}$$

$$y_{im} \geq 0 \qquad (i,m) \in A | I_i = -1 \tag{A.42}$$

$$s^S_{im} \quad \text{free} \qquad (i,m) \in A \tag{A.43}$$

Appendix B

Source Code

The mathematical model implemented in Xpress starts on the next page.

```
! Mathematical model
! Created by Marte Tuverud Kamphus, Spring 2017
! Norwegian University of Science and Technology


model Inventory_Perishable

options explterm
options noimplicit
uses  "mmxprs";
uses "mmsystem";


parameters
   DataFile= "Data_inventory.txt";
   RUNTIME = 50000;
end-parameters

setparam("XPRS_maxtime", RUNTIME);



!-------------------------------------------------------------------------
                    !Decleration of indices and sets
!-------------------------------------------------------------------------
declarations
    Ports:                  set of integer;
    Ships:                  set of integer;
    VisitM:                 dynamic array(integer) of set of integer;
    Destinations:           set of integer;

end-declarations

declarations
    nPorts:                 integer;
    nShips:                 integer;
    nMaxVisit:              integer;
    nVisitM:                dynamic array(integer) of integer;
end-declarations


initializations from DataFile
    nPorts;
    nShips;
    nMaxVisit;
end-initializations


Ports := 1 .. nPorts;                   !Set of all ports
Ships := 1 .. nShips;                   !Set of all ships
Destinations := 0 .. nPorts;            !Set of all ports + o(v)/d(v)


!creating visit number for ports
forall(ii in Ports, mm in 1..nMaxVisit ) do
    VisitM(ii) += {mm};
end-do

!creating visit number for o(v) and d(v)
forall(ii in Destinations, mm in 1..nShips | ii=0 ) do
    VisitM(0) += {mm};
end-do


finalize(Ports);
finalize(Ships);
finalize(Destinations);
```

```
!--------------------------------------------------------------------------
                    !Declearing parameters
!--------------------------------------------------------------------------

declarations

    SailingCost:                array(Destinations, Destinations)   of real;
    CapacityShip:               array(Ships)                        of real;
    UnitLoadTime:               array(Ports)                        of real;
    SailingDistance:            array(Destinations, Destinations)   of real;
    SailingSpeed:               array(Ships)                        of real;
    LengthPeriod:               real;
    PenaltySlaughterTime:       real;
    PenaltyDeliveredEnough:     array(Ports)                        of real;
    SafetyStockLower:           array(Ports)                        of real;
    PenaltyLower:               array(Ports)                        of real;
    Price:                      array(Ships)                        of real;
    MinTimePenalty:             real;
    MaxTime:                    real;
    InventoryInitial:           array(Ports)                        of real;
    InventoryMin:               array(Ports)                        of real;
    InventoryMax:               array(Ports)                        of real;
    WhichPort:                  array(Destinations)                 of integer;
    ProductionRate:             array(Ports)                        of real;
    ProdPort:                   array(Ports)                        of integer;
    ConPort:                    array(Ports)                        of integer;
end-declarations


initializations from DataFile
    SailingCost;
    CapacityShip;
    UnitLoadTime;
    SailingSpeed;
    SailingDistance;
    LengthPeriod;
    PenaltySlaughterTime;
    PenaltyDeliveredEnough;
    SafetyStockLower;
    PenaltyLower;
    Price;
    MinTimePenalty;
    MaxTime;
    InventoryInitial;
    InventoryMin;
    InventoryMax;
    WhichPort;
    ProductionRate;
    ProdPort;
    ConPort;

end-initializations


!-------------------------------------------------------------------------------
!                       Preprocessing
!-------------------------------------------------------------------------------
declarations
    SailingTime:        array(Destinations, Destinations, Ships)    of real;
end-declarations


forall(ii in Destinations, jj in Destinations, vv in Ships) do
    SailingTime(ii,jj,vv) :=  ceil((SailingDistance(ii,jj) / SailingSpeed(vv)));
end-do
```

```
!---------------------------------------------------------------------------------
!                            Declearing variables
!---------------------------------------------------------------------------------

declarations
    x:          dynamic array(Destinations, integer, Destinations, integer, Ships)
of mpvar;
    w:          dynamic array(Ports, integer)
of mpvar;

    tStart:     dynamic array(Ports, integer)
of mpvar;
    tEnd:       dynamic array(Ports, integer)
of mpvar;

    l:          dynamic array(Ports, integer, Ships)
of mpvar;
    q:          dynamic array(Ports, integer, Ships)
of mpvar;
    sStart:     dynamic array(Ports, integer)
of mpvar;
    sEnd:       dynamic array(Ports, integer)
of mpvar;

    safetyLow:  dynamic array(Ports, integer)
of mpvar;
    TimeSinceS: dynamic array(Ports, integer)
of mpvar;

    y:          dynamic array(Ports, integer)
of mpvar;
    slakk:      dynamic array(Ports, integer)
of mpvar;
end-declarations

!---------------------------------------------------------------------------------
!                        Creating variables
!---------------------------------------------------------------------------------


forall (ii in Destinations, mm in VisitM(ii), jj in Destinations, nn in VisitM(jj),
vv in Ships| (ii)<>(jj) or (ii=0 and jj=0) ) do
    if(jj=0) then
    create(x(ii,mm,0,vv,vv));
    x(ii,mm,0,vv,vv) is_binary;
    elif (ii=0) then
    create(x(0,vv,jj,nn,vv));
    x(0,vv,jj,nn,vv) is_binary;
    else
    create(x(ii, mm, jj, nn, vv));
        x(ii, mm, jj, nn, vv) is_binary;
    end-if
end-do

forall(ii in Ports, mm in VisitM(ii)) do
    create(w(ii,mm));
        w(ii,mm) is_binary;
end-do

forall (ii in Ports, mm in VisitM(ii)) do
    create(tStart(ii,mm));
end-do

forall (ii in Ports, mm in VisitM(ii)) do
    create(tEnd(ii,mm));
end-do
```

```
forall (ii in Ports, mm in VisitM(ii), vv in Ships) do
    create(l(ii,mm,vv));
        l(ii,mm,vv) is_integer;
end-do

forall (ii in Ports, mm in VisitM(ii), vv in Ships) do
    create(q(ii,mm,vv));
        q(ii,mm,vv) is_integer;
end-do

forall (ii in Ports, mm in VisitM(ii)) do
    create (sStart(ii,mm));
        sStart(ii,mm) is_free;
end-do

forall (ii in Ports, mm in VisitM(ii)) do
    create (sEnd(ii,mm));
end-do

forall (ii in Ports, mm in VisitM(ii)) do
    create (TimeSinceS(ii,mm));
end-do

forall (ii in Ports, mm in VisitM(ii)| WhichPort(ii) =-1) do
    create (y(ii,mm));
end-do

forall (ii in Ports, mm in VisitM(ii)) do
    create (safetyLow(ii,mm));
end-do


!-----------------------------------------------------------------------------------
--
!                       Declaration of objective function and constraints
!-----------------------------------------------------------------------------------
--


declarations
    TotalCost:                                                              linctr;

    Con1:               array(Ports, integer)                               of
linctr;

    Con23:              array(Ships)                                        of
linctr;
    Con233:             array(Ports, integer, Ships)                        of
linctr;

    Con4:               array(Ships)                                        of
linctr;
    Con5:               array(Ports, integer)                               of
linctr;
    ConInitialLoad:     array(Ships)                                        of
linctr;
    Con6a:              array(Ports, integer, Ports, integer, Ships)        of
linctr;
    Con6b:              array(Ports, integer, Ports, integer, Ships)        of
linctr;
    Con7:               array(Ports, integer, Ships)                        of
linctr;
    Con8:               array(Ports, integer, Ships)                        of
linctr;
    Con9:               array(Ports, integer, Ships)                        of
linctr;
```

```
    ConUnl:                  array(Ports, integer, Ships)                 of
linctr;

    ClearShip:               array(Ports, integer, Ports, integer, Ships)    of
linctr;
    ClearShip2:              array(Ports, integer, Ships)                 of
linctr;

    conCon:                  array(Ports, integer, Ports, integer, Ships)    of
linctr;
    Con11:                   array(Ports, integer)                         of
linctr;
    ConL2:                   array(Ports, integer)                         of
linctr;
    ConMaxTime:              array(Ports, integer)                         of
linctr;
    ConVisit1:               array(Ports, integer)                         of
linctr;
    ConVisit2:               array(Ports, integer)                         of
linctr;

    TimeSinceSlaughter:      array(Ports, integer, Ships)                 of
linctr;
    TimeSinceSlaughter2a:    array(Ports, integer, Ports, integer, Ships)    of
linctr;
    TimeSinceSlaughter2b:    array(Ports, integer, Ports, integer, Ships)    of
linctr;


    MaxTimeSinceS:           array(Ports, integer)                         of
linctr;

    yPen:                    array(Ports, integer)                         of
linctr;
    yPen1:                   array(Ports, integer)                         of
linctr;

    oneto1b:                 array(Ports, integer, Ports, integer, Ships)    of
linctr;
    oneto1c:                 array(Ports, integer, Ports, integer, Ships)    of
linctr;


    s1:                      array(Ports, integer)                         of
linctr;
    s2:                      array(Ports, integer)                         of
linctr;
    s3:                      array(Ports, integer)                         of
linctr;
    s32:                     array(Ports, integer)                         of
linctr;
    upper1:                  array(Ports, integer)                         of
linctr;
    upper2:                  array(Ports, integer)                         of
linctr;


    Lower1:                  array(Ports, integer)                         of
linctr;
    Lower2:                  array(Ports, integer)                         of
linctr;

    safetylower:             array(Ports, integer)                         of
linctr;
    Con14a:                  array(Ports, integer)                         of
linctr;
    Con14b:                  array(Ports, integer)                         of
linctr;
```

```
    Con14a2:                array(Ports, integer)                          of
linctr;
    Con14b2:                array(Ports, integer)                          of
linctr;

    InitialLoad:            array(Destinations, integer, Ships)            of
linctr;
    init:                   array(Ships)                                   of
linctr;
end-declarations

!--------------------------------------------------------------------------------
---
!                       Formulations of objective function and contraints
!--------------------------------------------------------------------------------
---
!A.X refers the constraint number in Appendix A



TotalCost :=
  (sum(ii in Ports, mm in VisitM(ii), jj in Ports, nn in VisitM(jj), vv in Ships)
SailingCost(ii, jj)*x(ii,mm,jj,nn,vv))
+ (sum(jj in Ports, nn in VisitM(jj), vv in Ships) Price(vv)* x(0,vv,jj,nn,vv))
+ (sum(ii in Ports, mm in VisitM(ii)) PenaltyLower(ii)*safetyLow(ii,mm))
+ (sum(ii in Ports) PenaltyDeliveredEnough(ii)*(1- ( (sum(mm in VisitM(ii), vv in
Ships) q(ii,mm,vv)) /(ProductionRate(ii)*LengthPeriod + InventoryInitial(ii)))))
+ (sum(ii in Ports, mm in VisitM(ii)) PenaltySlaughterTime*y(ii,mm));


!----------FLOW CONSTRAINTS-------------------------------------------------------
----

!A.4
forall(ii in Ports, mm in VisitM(ii)) do
    Con1(ii,mm) :=
        (sum(jj in Destinations, nn in VisitM(jj), vv in Ships) x(ii,mm,jj,nn,vv))
+ w(ii,mm)
        = 1;
end-do

!A.5
forall(vv in Ships) do
        Con23(vv) :=
        sum(jj in Ports, nn in VisitM(jj)| (WhichPort(jj)=1 )) x(0,vv,jj,nn,vv) +
x(0,vv,0,vv,vv) = 1;
end-do

!A.6
forall(ii in Ports, mm in VisitM(ii), vv in Ships) do
        Con233(ii,mm,vv) :=
        sum(jj in Ports, nn in VisitM(jj)) x(jj,nn,ii,mm,vv) + x(0,vv,ii,mm,vv) -
sum(jj in Ports, nn in VisitM(jj)) x(ii,mm,jj,nn,vv) - x(ii,mm,0,vv,vv)= 0;
end-do

!A.7
forall(vv in Ships) do
    Con4(vv) :=
        (sum(ii in Ports, mm in VisitM(ii))x(ii,mm,0,vv,vv)) + x(0,vv,0,vv,vv)
        = 1;
end-do

!A.8
forall(ii in Ports, mm in VisitM(ii)| mm>1) do
    Con5(ii,mm) :=
        w(ii,mm) - w(ii,mm-1)
        >= 0;
end-do
```

```
!---------LOAD/UNLOAD CONSTRAINTS---------------------------------------------------
-------------



!A.9 AND A.10
forall(ii in Ports, mm in VisitM(ii), jj in Ports, nn in VisitM(jj), vv in Ships)
do

    Con6a(ii,mm,jj,nn,vv) :=
        l(ii,mm,vv) + WhichPort(jj)*q(jj,nn,vv) - l(jj,nn,vv) +
CapacityShip(vv)*x(ii,mm,jj,nn,vv)
        <= CapacityShip(vv);

    Con6b(ii,mm,jj,nn,vv) :=
        l(ii,mm,vv) + WhichPort(jj)*q(jj,nn,vv) - l(jj,nn,vv) -
CapacityShip(vv)*x(ii,mm,jj,nn,vv)
        >= - CapacityShip(vv);
end-do


!A.11
forall(ii in Ports, mm in VisitM(ii), vv in Ships | WhichPort(ii)= 1) do
    Con7(ii,mm,vv) :=
        q(ii,mm,vv) <= l(ii,mm,vv);

    Con8(ii,mm,vv) :=
        l(ii,mm,vv) <= (sum(jj in Destinations, nn in VisitM(jj))
CapacityShip(vv)*x(ii,mm,jj,nn,vv));
end-do


!A.12
forall(ii in Ports, mm in VisitM(ii), vv in Ships | WhichPort(ii)= -1) do
    Con9(ii,mm,vv) :=
        l(ii,mm,vv) <= (sum(jj in Destinations, nn in VisitM(jj))
CapacityShip(vv)*x(ii,mm,jj,nn,vv)) - q(ii,mm,vv);
end-do



!A.32
forall(ii in Ports, mm in VisitM(ii), vv in Ships | WhichPort(ii) =-1) do
    ConUnl(ii,mm,vv) :=
        q(ii,mm,vv) <= sum(jj in Destinations, nn in VisitM(jj))
(InventoryMax(ii)*x(ii,mm,jj,nn,vv));
end-do


!A.13
forall(ii in Ports, mm in VisitM(ii), jj in Ports, nn in VisitM(jj), vv in Ships |
WhichPort(ii) = -1 and WhichPort(jj) = 1) do
    ClearShip(ii,mm,jj,nn,vv) :=
        l(ii,mm,vv) <= CapacityShip(vv)*(1-x(ii,mm,jj,nn,vv)) ;
end-do

!A.14
forall(ii in Ports, mm in VisitM(ii), vv in Ships ) do
    ClearShip2(ii,mm, vv) :=
        l(ii,mm,vv) <= CapacityShip(vv)*(1-x(ii,mm,0,vv,vv)) ;
end-do


!-------------------------TIME CONSTRAINTS-----------------------------------------
--
```

```
!A.15
forall(ii in Ports, mm in VisitM(ii), jj in Ports, nn in VisitM(jj), vv in Ships |
ii<>jj ) do
    conCon(ii, mm, jj, nn, vv) :=
        tEnd(ii,mm) + SailingTime(ii,jj,vv)*x(ii,mm,jj,nn,vv) - tStart(jj,nn) +
LengthPeriod*x(ii,mm,jj,nn,vv) <= LengthPeriod;
end-do


!A.16
forall(ii in Ports, mm in VisitM(ii)) do
    ConL2 (ii,mm) :=
        tStart(ii,mm) + sum(vv in Ships)(q(ii,mm,vv)/UnitLoadTime(ii)) =
tEnd(ii,mm);
end-do

!A.17
forall(ii in Ports, mm in VisitM(ii)) do
    ConMaxTime(ii,mm) :=
        tEnd(ii,mm) <= LengthPeriod;
end-do

!A.18
forall(ii in Ports, mm in VisitM(ii)|(mm)>1) do
    ConVisit1(ii,mm) :=
        tStart(ii,mm)- tEnd(ii,mm-1) + LengthPeriod*w(ii,mm) <= LengthPeriod;

end-do

!A.19
forall(ii in Ports, mm in VisitM(ii)| mm <> 1) do
    Con11(ii,mm) :=
        tStart(ii,mm) - tEnd(ii,mm-1) >= 0;
end-do



!--------------------PERISHABLE CONSTRAINTS--------------------------------
!A.20
forall(ii in Ports, mm in VisitM(ii), vv in Ships | WhichPort(ii) = 1) do
    TimeSinceSlaughter (ii,mm,vv) :=
        TimeSinceS(ii,mm) >= (sStart(ii,mm)/ProductionRate(ii)) +
q(ii,mm,vv)/UnitLoadTime(ii);
end-do


!A.21 AND A.22
forall(ii in Ports, mm in VisitM(ii), jj in Ports, nn in VisitM(jj), vv in Ships |
WhichPort(ii) = 1 and WhichPort(jj)= 1) do

    oneto1b(ii,mm,jj,nn,vv) :=
        TimeSinceS(jj,nn) + LengthPeriod >= sStart(jj,nn)/ProductionRate(jj) +
q(jj,nn,vv)/UnitLoadTime(jj) + LengthPeriod*x(ii,mm,jj,nn,vv);

    oneto1c(ii,mm,jj,nn,vv) :=
        TimeSinceS(jj,nn) + LengthPeriod >= TimeSinceS(ii,mm) + tEnd(jj,nn) -
tEnd(ii,mm) + LengthPeriod*x(ii,mm,jj,nn,vv);
end-do

!A.23 AND A.24
forall(ii in Ports, mm in VisitM(ii), jj in Ports, nn in VisitM(jj), vv in Ships |
WhichPort(jj)=-1) do

    TimeSinceSlaughter2a (ii,mm,jj,nn,vv) :=
        TimeSinceS(ii,mm) + tEnd(jj,nn) - tEnd(ii,mm) -TimeSinceS(jj,nn)
+LengthPeriod*x(ii,mm,jj,nn,vv)
        <= LengthPeriod;
```

```
    TimeSinceSlaughter2b (ii,mm,jj,nn,vv) :=
        TimeSinceS(ii,mm) + tEnd(jj,nn) - tEnd(ii,mm) -TimeSinceS(jj,nn) -
LengthPeriod*x(ii,mm,jj,nn,vv)
        >= - LengthPeriod;
end-do

!A.25
forall(ii in Ports, mm in VisitM(ii)) do
    MaxTimeSinceS (ii,mm) :=
        TimeSinceS(ii,mm) <= MaxTime;
end-do


!A.26
forall(ii in Ports, mm in VisitM(ii)|WhichPort(ii)=-1) do
    yPen(ii,mm) :=
    y(ii,mm) >= TimeSinceS(ii,mm) - MinTimePenalty;
end-do



!------------ INVENTORY CONSTRAINTS----------------------------------------


!A.27 AND A.28
forall(ii in Ports,mm in VisitM(ii)) do
    if (mm=1) then
        s2(ii,mm):=
        InventoryInitial(ii) +  ProductionRate(ii)*tStart(ii,mm) = sStart(ii,mm);
        else
        s3(ii,mm):=
        sEnd(ii, mm-1) +  ProductionRate(ii)*(tStart(ii,mm)-tEnd(ii, mm-1)) <=
sStart(ii,mm);
        s32(ii,mm):=
        sEnd(ii, mm-1) +  ProductionRate(ii)*(tStart(ii,mm)-tEnd(ii, mm-1)) >=
sStart(ii,mm);

    end-if
end-do

!A.29
forall (ii in Ports, mm in VisitM(ii)) do
    s1(ii,mm):=
    sStart(ii,mm) +  ProductionRate(ii)*(tEnd(ii,mm)-tStart(ii,mm))-
WhichPort(ii)*(sum(vv in Ships) q(ii,mm,vv))
    = sEnd(ii,mm);
end-do


!A.30 AND A.35
forall (ii in Ports, mm in VisitM(ii)| WhichPort(ii) = 1) do
    if(mm=nMaxVisit) then
        upper1(ii,mm):=
        sEnd(ii,mm)+ProductionRate(ii)*(LengthPeriod-
tEnd(ii,mm))<=InventoryMax(ii);
    end-if
        upper2(ii,mm):=
        sStart(ii,mm) <= InventoryMax(ii);
end-do

!A.31 AND A.37
forall (ii in Ports, mm in VisitM(ii)| WhichPort(ii) = -1) do
    if(mm=nMaxVisit) then
        Lower1(ii,mm):=
        sEnd(ii,mm)+ProductionRate(ii)*(LengthPeriod-tEnd(ii,mm)) >=
SafetyStockLower(ii);
    end-if
        Lower2(ii,mm):=
```

```
        sEnd(ii,mm) <= InventoryMax(ii);
end-do

!A.34
forall(ii in Ports, mm in VisitM(ii) | WhichPort(ii) = 1) do
    Con14a(ii,mm) :=
        InventoryMin(ii) <= sEnd(ii,mm);
end-do


!A.36
forall(ii in Ports, mm in VisitM(ii) | WhichPort(ii) = -1) do
    Con14a2(ii,mm) :=
        InventoryMin(ii) <= sStart(ii,mm);
end-do

!A.33
forall(ii in Ports, mm in VisitM(ii) | WhichPort(ii) = -1) do
    safetylower(ii,mm) :=
        sStart(ii,mm) + safetyLow(ii,mm) >= SafetyStockLower(ii);
end-do



! --------------------------------------------------------------------------------
---------
!                           Investigate parameters
! --------------------------------------------------------------------------------
---------

setparam("XPRS_VERBOSE", TRUE);

declarations
                status:      string;
end-declarations

case getprobstat of
                XPRS_OPT: status := "Optimal";
                XPRS_UNF: status := "Unfinished";
                XPRS_INF: status := "Infeasible";
                XPRS_UNB: status := "Unbounded";
                XPRS_OTH: status := "Failed";
                else      status := "Unknown";
end-case

! --------------------------------------------------------------------------------
-------
!                           Minimize objective function
! --------------------------------------------------------------------------------
-------
minimize(TotalCost);


!---------------------------------------------------------------------------------
------
!                           Creating of output file
! --------------------------------------------------------------------------------
-------

declarations
    Outputname: string;
end-declarations


Outputname := 'Output_' + DataFile + '.txt';
fopen(Outputname,F_OUTPUT);
        writeln;
        writeln("Best solution: "+getsol(TotalCost));
```

```
        writeln("Best bound: "+strfmt(getparam("xprs_bestbound"),12,2));
        writeln("Gap :"+(getsol(TotalCost)-
getparam("xprs_bestbound"))/getsol(TotalCost));
        writeln("Number rows(orig): ", strfmt(getparam('xprs_originalrows'),12));
        writeln("Number col(orig):  ", strfmt(getparam('xprs_originalcols'),12));
        writeln("Number rows(pre):  ", strfmt(getparam('xprs_rows'),12));
        writeln("Number col(pre):   ", strfmt(getparam('xprs_cols'),12));
        writeln("Number nodes :     ", strfmt(getparam("xprs_nodes"),12));

        writeln;

    forall(vv in Ships) do
        forall(ii in Ports) do
            forall(mm in VisitM(ii)) do
                forall(jj in Destinations) do
                    forall(nn in VisitM(jj)) do
                        if (getsol(x(ii,mm,jj,nn,vv))=1) then
                            write("Ship "+vv+" loads/unloads " +
getsol(q(ii,mm,vv))+ " tons of salmon at (" + ii+"," +mm+
                                 ") and sails to (" +jj+ "," +nn+ ") with " +
getsol(l(ii,mm,vv)) +" tons of salmon onboard. Service started at "
+getsol(tStart(ii,mm))+ " and ended at " + getsol(tEnd(ii,mm))+ ". Stock after
service at (" +ii+","+mm+ ") is " + getsol(sEnd(ii,mm)));
                            writeln;
                        end-if
                    end-do
                end-do
            end-do
        end-do
    end-do
    writeln;

    forall(ii in Destinations, mm in VisitM(ii), jj in Destinations, nn in
VisitM(jj), vv in Ships | WhichPort(jj)=-1) do
        if(getsol(x(ii,mm,jj,nn,vv))=1) then
            write( "Ship " + vv + " unloads " + getsol(q(jj,nn,vv)) + " tons of
salmon in port( " + jj + "," + nn + " ). The salmon is "
+getsol(TimeSinceS(jj,nn))+ " hours old when being delivered");
        writeln;
        end-if
    end-do
    writeln;
    writeln("Transportation costs are: " + sum(ii in Ports, mm in VisitM(ii), jj in
Ports, nn in VisitM(jj), vv in Ships)
SailingCost(ii,jj)*getsol(x(ii,mm,jj,nn,vv)));
    writeln("Penalty lower stock: " +  + sum(ii in Ports, mm in VisitM(ii))
PenaltyLower(ii)*getsol(safetyLow(ii,mm)));
    writeln("Penalty delivered enough: " +  sum(ii in Ports) (1- ( (sum(mm in
VisitM(ii), vv in Ships) (getsol(q(ii,mm,vv))))  /(ProductionRate(ii)*LengthPeriod
+ InventoryInitial(ii))))*PenaltyDeliveredEnough(ii));
    writeln("Penalty for late delivery: " + sum(ii in Ports, mm in VisitM(ii))
getsol(y(ii,mm))*PenaltySlaughterTime);
    writeln("Number of vessels used: " +sum(jj in Ports, nn in VisitM(jj), vv in
Ships) (getsol(x(0,vv,jj,nn,vv))));

    writeln;
    writeln("Delivered in the period is: " + sum(ii in Ports, mm in VisitM(ii), vv
in Ships) (getsol(q(ii,mm,vv)))*ConPort(ii));
    writeln("Produced in the period is: " + sum(ii in
Ports)(ProductionRate(ii)*LengthPeriod + InventoryInitial(ii))*ProdPort(ii));
    writeln("delivered vs produced equal: " + (sum(ii in Ports, mm in VisitM(ii),
vv in Ships)( getsol(q(ii,mm,vv)))*ProdPort(ii))/(sum(ii in
Ports)(ProductionRate(ii)*LengthPeriod + InventoryInitial(ii))*ProdPort(ii)));


writeln("-----------------------------------------------------------------
----------");
```

```
writeln(strfmt('Name',8), strfmt('Activity',19), strfmt('Reduced cost',
19),strfmt('Coefficient',15));
writeln("--------------------------------------------------------------------------
----------");

forall (ii in Destinations, mm in VisitM(ii), jj in Destinations, nn in VisitM(jj),
vv in Ships| getsol(x(ii,mm,jj,nn,vv))>0.1) do
    write(strfmt('x(' + ii + ',' + mm +',' + jj + ',' + nn+ ',' + vv + ')',8));
    write(strfmt(getsol(x(ii,mm,jj,nn,vv)),19));
    write(strfmt(getrcost(x(ii,mm,jj,nn,vv)),19,2));
    write(strfmt(getcoeff(TotalCost,x(ii,mm,jj,nn,vv)),15));
    writeln;
end-do
writeln;
fclose(F_OUTPUT);



end-model
```

Appendix C

Input File for Scenario 1

```
nPorts : 2
nShips : 2
nMaxVisit : 13
LengthPeriod : 500

SailingSpeed : [ 15 15 ]
CapacityShip : [ 2500 2000 ]
Price : [ 900 800 ]  !*1000

ProductionRate : [ 35 -35]
WhichPort : [0  1  -1 ]

InventoryInitial : [2500 3000 ]
InventoryMin : [ 0 -120 ]
InventoryMax : [ 3000 4000]
SafetyStockLower: [0 0 ]


!Penalty cost *1000
PenaltySlaughterTime :  [1]
PenaltyDeliveredEnough : [500 0]
PenaltyLower :      [ 0 2 ]


MinTimePenalty : 96
MaxTime : 144


ProdPort : [ 1 0]
ConPort : [0 1]

UnitLoadTime : [ 475 475 ]


SailingDistance :  [ 0 0 0
    0 0 685
    0 685  0
    ]

!*1000
SailingCost : [ 0 0 0
    0 0 105
    0 105 0
    ]
```

Appendix D

Input File for Scenario 2

```
nPorts : 3
nShips : 4
nMaxVisit : 15


CapacityShip : [ 2500 2500 2500 2000]
SailingSpeed : [ 15 15 15 15]
Price : [ 900 900 900 800 ]

ProductionRate : [40 40 -80]
WhichPort : [0 1  1 -1]

InventoryInitial : [2500 2000 5800 ]
InventoryMin : [ 0 0 -120 ]
InventoryMax : [ 5000 5000  10000]
LengthPeriod : 336 ! 2 weeks

ProdPort : [ 1 1 0]
ConPort : [ 0 0 1]

PenaltySlaughterTime :  [ 1 ]
PenaltyDeliveredEnough :    [500 500 0 ]
PenaltyLower :       [0 0 2 ]
MinTimePenalty : 96
MaxTime : 144
SafetyStockLower: [0 0 0 ]

UnitLoadTime : [ 475 475 475]

!Hitra-Rørvik-Cuxhaven
SailingDistance :  [ 0 0 0 0
    0 0 98  767
    0 98 0 685
    0 767 685 0 ]

!*1000
SailingCost : [ 0 0 0 0
    0  0 15 118
    0 15 0 105
    0 118 105 0
    ]
```