NTNU

Norwegian University of
Science and Technology

Designing Feeder Routes in a Mother-Daughter Vessel System with Ship-to-Ship Cargo Transfer

A Combined Optimization-Simulation
Approach

Hans Henrik Nordheim

Marine Technology
Submission date: June 2017
Supervisor: Bjørn Egil Asbjørnslett, IMT

Norwegian University of Science and Technology
Department of Marine Technology

Preface

This thesis is the result of my work during the final semester at the Norwegian University of Science and Technology to receive the grade Master of Science in Marine Technology. The scope of the thesis is 30 credits, which corresponds to the workload of a full semester.

I would like to thank my supervisor, Bjørn Egil Asbjørnslett, for help and guidance throughout the process. I would also like to thank Kay Fjørtoft from SINTEF Ocean for keeping me updated on the Short Sea Pioneer project and Ann Iren Holm Rise from Trondheim Port Authority for providing me with useful information about port operations.

Trondheim, 09.06.2017

Hans H. Nordheim

Hans Henrik Nordheim

Summary

This thesis aims to develop decision support models for designing daughter vessel feeder routes for the mother-daughter vessel system Short Sea Pioneer. Short Sea Pioneer is a next generation short sea shipping system that utilizes daughter vessels to feed containers between small and remote ports and the mother vessels. Although Short Sea Pioneer presents potential benefits such as shorter roundtrip times and reduced time spent in port for the mother vessels, the system also poses higher complexity compared to conventional short sea shipping systems by introducing more elements that need to be precisely coordinated. Cargo transfer between mother and daughter vessels finds place at sea which implies that the schedules for the mother vessels and daughter vessels must be coordinated.

An optimization model is constructed to generate daughter vessel feeder routes based on required number of visits for the involved ports for a given planning horizon and the scheduled arrival times of mother vessels in the area. To assess and increase the robustness of the generated routes, and to ensure that the system can cope with small variations in the operating environment, a discrete event simulation model is developed. The simulation model is used to evaluate the generated routes when the system is subject to uncertainties and a dynamic nature. A script to estimate the operability of the ship-to-ship cargo transfer operation based on operational criterion and the duration of the operation is also made. The constructed models are limited to the operation of one daughter vessel.

To demonstrate potential application areas of the models, and to verify that they work as intended, the models are applied on a constructed feeder case for the Trondheim Fjord. The simulation model is verified by first running it with deterministic input data. The effect of 24/7 port service versus restricted port opening hours is evaluated. The analysis shows that results from the optimization model are not much affected by introducing restricted port opening hours. However, when the routes are evaluated in the simulation model, it is evident that the robustness of the routes decrease for restricted port opening hours. By designing the routes for transportation demands above the average level, new routes are established. These routes perform better in the stochastic scenario. Combined use of the optimization and simulation model highlights the importance of considering uncertainties when designing the system.

The constructed models are based on a range of assumptions about the operating context of the vessel that should be kept in mind if one wishes to utilize the models. Regarding further work it is recommended to include stakeholders in the project and system experts to review and revise the assumptions to increase the credibility of the models. Stakeholders should also be included in the formulation of the objective functions to make sure that their requirements are better met in the solutions.

Sammendrag

Denne oppgaven tar sikte på å lage beslutningsstøttemodeller for design av ruter for datterfartøy i mor-datterfartøysystemet Short Sea Pioneer. Short Sea Pioneer er et innovativt nærskipsfartsystem som utnytter datterfartøy til å frakte containere mellom små og avsidesliggende havner og morfartøyene. Selv om Short Sea Pioneer byr på potensielle fordeler som kortere rundturtid og redusert tidsbruk i havn for morfartøyene innebærer også systemet høyere kompleksitet sammenlignet med konvensjonelle nærskipsfartløsninger ved å introdusere flere elementer som må koordineres. Lastoverføring mellom mor og datterfartøy vil gjennomføres til sjøs, noe som betyr at rutene for mor og datterfartøy må samkjøres.

En optimeringsmodell er laget for å generere datterfartøyruter basert på nødvendig antall besøk for de involverte havnene og planlagte ankomsttider av morfartøy i området. For å vurdere og øke robustheten til de genererte rutene, og for å sikre at systemet kan håndtere små variasjoner i miljøet, utvikles en diskret hendelsessimuleringsmodell. Simuleringsmodellen brukes for å teste rutene når systemet utsettes for usikkerhet og variasjoner. Et script for å estimere operabilitet av lastoverføringsoperasjonen mellom skipene basert på operasjonskriterium og varighet av operasjonen er også laget. Modellene er begrenset til operasjon av ett datterfartøy.

For å demonstrere potensielle bruksområder av modellene, og for å verifisere at de virker som tiltenkt, testes modellene på en konstruert case for Trondheimsfjorden. Simuleringsmodellen verifiseres ved å først kjøre modellen med deterministisk input data og sammenlikne resultatene med optimeringsmodellen. Effekten av døgnåpne havner kontra begrensede åpningstider evalueres. Analysene viser at resultatene fra optimeringsmodellen er lite påvirket av begrensede åpningstider. Når rutene testes i simuleringsmodellen er det derimot tydelig at robustheten av rutene faller når begrensede åpningstider innføres. Ved å designe rutene for transportbehov over gjennomsnittsnivået etableres nye ruter. Disse rutene fungerer bedre i det stokastiske scenariet. Kombinert bruk av optimering og simuleringsmodellen belyser viktigheten av å vurdere usikkerhet ved utforming av systemet.

Modellene er basert på en rekke antagelser om operasjonsmønsteret til fartøyet. Disse er det viktig å ta hensyn til om man ønsker å utnytte modellene. I videre arbeid anbefales det å inkludere interessenter i prosjektet og systemeksperter for å gjennomgå og revidere antakelsene for å øke troverdigheten av modellene. Interessenter bør også inkluderes i formuleringen av objektivfunksjonene for å sikre at deres ønsker bedre tas hensyn til i løsningene.

Table of Contents

List of Figures

List of Tables

1. Introduction

Background

There is a high political pressure to transfer goods from road to sea transportation. The EU have set a clear and ambitious goal of shifting 30% of the goods that today are transported by trucks on distances longer than 300 km to rail or sea transportation by 2020. The corresponding goal for 2050 is 50% (European Commission. Directorate-General for Mobility, 2011). The Norwegian government have also announced that they will support measures to increase the competitiveness of short sea shipping solutions (Samferdselsdepartement, 2013).

Institute of Transport Economics (2015) state that despite an increased focus on achieving a modal shift from road to sea and rail, actual measures having a good effect have failed to appear so far. For the modal shift to find place they stress that port costs will need to be reduced and that the departures will need to find place more often.

Short Sea Pioneer is a next generation short sea container shipping system that is currently under development by NCE Maritime CleanTech. The system is developed to solve some of the challenges currently faced by the short sea shipping industry and consists of two different types of vessels: mother vessels as distributors and daughter vessels as feeders. The mother vessels will be used to transport containers on relatively long distances and only visit larger and easily accessible ports along it's roundtrips. The daughter vessels will be used in areas where there is a cluster of ports, and will be used to feed containers from the ports in the area to the mother vessels, and vice versa. Cargo transfer between mother and daughter vessels will find place at sea.

By utilizing daughter vessels to feed containers one can access a larger number of ports and at the same time decrease the number of port calls for the large mother vessels. Hence it is believed that the mother-daughter vessel system will decrease the total roundtrip time which in turn can allow for more frequent departures. Using daughter vessels as feeders also allows for utilizing economics of scale by increasing the size of the mother vessels as ports with draft limitations will be serviced by a daughter vessel.

Although the mother-daughter vessel system Short Sea Pioneer presents potential benefits, it also poses a higher level of complexity by adding more elements to the system. As Short Sea Pioneer will operate as a liner shipping system the vessels will need to follow fixed pre-published schedules. The mother vessel system and the daughter vessel system will therefore need to be precisely coordinated.

Objective

The objective of this thesis is to develop decision support models for designing daughter vessel feeder routes in the mother-daughter vessel system Short Sea Pioneer to ensure that each port is visited its desired number of times and that the daughter vessel do not delay the mother vessels.

The models will be applied on a constructed feeder case for the Trondheim area to demonstrate potential application areas of the models.

Structure of the report

The remainder of this thesis is organized as follows: Chapter 2 presents the main characteristics of the Short Sea Pioneer system while chapter 3 presents the part of the system that is analyzed in this thesis, limitations and assumptions. In chapter 4, relevant literature is presented related to existing transportation systems with similarities to Short Sea Pioneer and modelling of maritime transportation systems. The methodology used to address the research questions from chapter 3 are described in chapter 5, while the models constructed based on the chosen methodology are described in chapter 6. A computational study is presented in chapter 7, where the models are applied on a constructed feeder case for the Trondheim Fjord. The work is discussed in chapter 8. Concluding remarks are given in chapter 9, while recommendations for further work are provided in chapter 10.

2. Short sea pioneer system description

Short Sea Pioneer is an unconventional container shipping system that consists of two different types of vessels: mother vessels as distributors and daughter vessels as feeders. The mother vessels will travel a roundtrip where they only visit relatively large and easily accessible ports along the route. They will also meet up with a daughter vessel at designated points along the route to perform cargo transfer between the vessels at sea. The daughter vessels will be used in areas where there is a cluster of ports and/or where the ports are located at a considerable distance from the main sailing route for the mother vessel, and will be used to feed containers between these ports and the mother vessels.

A simplified example of a conventional short sea shipping system operating on diversified route is illustrated in Figure 1. The large black nodes illustrate relatively large ports located close to the vessel's main sailing route while the smaller black nodes illustrate smaller ports located at a larger distance from the main sailing route, typically in a fjord.

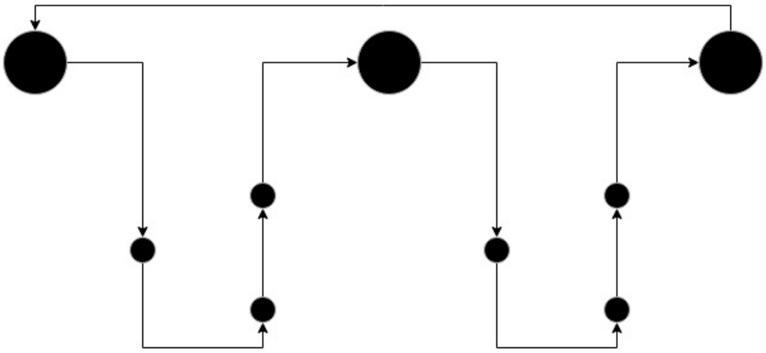


Figure 1: Conventional short sea liner shipping route. One vessel serving all ports that are part of the trade.

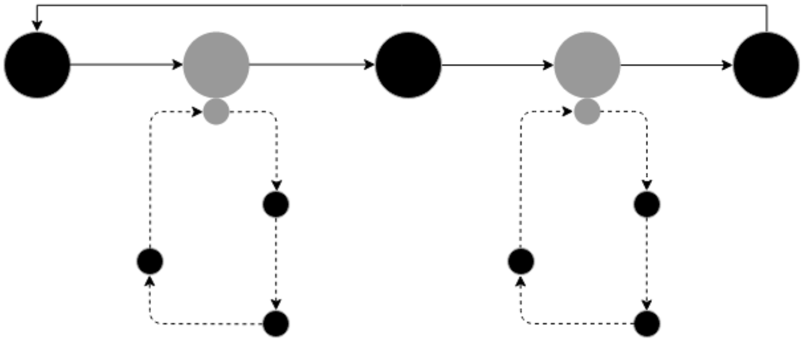


Figure 2: Logistics split obtained by Short Sea Pioneer. Route for daughter vessels illustrated by dotted lines. Daughter vessels feed containers between small and remote ports and the mother vessels.

For the conventional short sea shipping route, the same ship is used to serve both large and small ports along it's roundtrip, which results in many port calls and considerable deviations from the main sailing route for the large vessels.

The main principle of the Short Sea Pioneer is illustrated in Figure 2, where a logistics split is obtained between the large and easily accessible ports, and the smaller and more remote ports. The route for the mother vessels are illustrated by solid lines while the route for the daughter vessels are illustrated by dotted lines. The grey nodes in Figure 2 illustrate areas at sea where cargo transfer will be carried out between mother- and daughter vessel. For this operation to find place, naturally both a mother vessel and a daughter vessel will need to be present at this area. Hence, the grey nodes represent areas where the mother vessel system and the daughter vessel system interfere.

By dividing the transportation service between mother and daughter vessels, the mother vessels will decrease their number of port calls and deviations from their main sailing route, and hence reduce their total roundtrip time. Limiting the ports serviced by the mother vessels to larger ports also allows for utilizing economics of scale by increasing the size of the mother vessels as smaller ports with low port infrastructure and shallow water depth will be serviced by a daughter vessel. Using self-sufficient daughter vessels equipped with cranes also allows for expanding the trade-route to several small ports without the need for duplicate resources in ports, such as quay cranes and deep-water berths.

To safely and efficiently perform cargo transfer between mother vessels and daughter vessels, this operation will find place in relatively sheltered areas, for instance in fjord inlets. All cargo transfer between the vessels will find place when the vessels lie still. Containers will be piled and handled in racks between mother and daughter vessel to improve cargo transfer rates compared to port operations by crane.

Short Sea Pioneer is designed to operate as a liner shipping system and will therefore follow fixed pre-published schedules. The schedules should further be cyclical to provide a predictable service for customers over time.

3. Problem description and limitations

The mother-daughter vessel system Short Sea Pioneer presents potential benefits such as shorter roundtrip times, reduced time in ports for the mother vessels, and access to smaller ports and ports with draft limitations. However, using daughter vessels to distribute cargo in parts of the shipping system increase the complexity of the system by introducing more elements that need to be precisely coordinated.

This thesis aims to develop an optimization model and a discrete event simulation model as decision support tools to determine daughter vessel sailing speed and feeder routes to ensure sufficient frequency of visits in ports without delaying the mother vessels. A script to analyze historical wave data is also made to estimate the operability of the ship-to-ship cargo transfer operation. The optimization model is configured to find daughter vessel minimum speed and routes, while the simulation model is used to evaluate and adjust the solution, when the system is subject to uncertainties, to ensure a more robust solution. The models are limited to the operation of one daughter vessel.

The planning horizon is divided into a number of periods, where the end of a period represents the arrival of a mother vessel. Feeder routes are designed for each period and are configured to be cyclical, meaning that when the route for the last period is completed, the vessel again starts sailing the route for the first period.

The main priority when designing the daughter vessel feeder routes is to generate routes that can be travelled with the lowest possible speed, as speed is assumed to be the main cost-driving factor. The second priority is to minimize the total distance sailed by the daughter vessel.

The mother vessels are treated as the governing part of the system as a delayed mother vessel can result in delayed deliveries in all parts of the system, which in turn can result in added cost and loss of customers. It is therefore vital that a daughter vessel is present when a mother vessel arrives at the transshipment meeting point. Mother vessel arrivals are modelled as fixed points in time in the models, and hence it is assumed that the mother vessels always arrive on time.

The ship-to-ship cargo transfer operation between mother and daughter vessel is limited by a certain significant wave height. If the weather conditions are too severe to carry out the ship-to-ship cargo transfer operation at sea, the vessels will sail to a nearby port and perform the cargo transfer in shelter. It is assumed that there is no available storage for containers in this port, meaning that the daughter vessel must wait for the mother vessel before it can start (un)loading containers even if the cargo transfer finds place in port.

Because meetings between mother vessel and daughter vessels are relatively spread in time, it is assumed that significant wave height at the transshipment meeting point is independent of significant wave height experienced at the previous meeting. Cargo transfer between mother and daughter vessel also finds place in relatively sheltered areas where wind generated waves are more dominant.

It should be noted that no commodity flow analysis is carried out in this thesis. The thesis aims to develop decision support models that can be used for different areas and with different transportation volumes. It is further assumed perfect balance between ingoing and outgoing containers for each port visit. That is, every time a port is visited the number of containers loaded is equal to the number of containers unloaded.

The models are applied on a constructed feeder problem for the Trondheim Fjord to demonstrate potential application areas of the models, and to verify that they work as intended. The effect of restricted opening hours versus 24/7 service in the ports is evaluated.

4. Literature review

This chapter provides an overview of relevant literature related to maritime transportation systems with similarities to Short Sea Pioneer, and modelling of maritime transportation systems. The literature study is mainly carried out by use of Oria, NTNU's online library.

4.1 Maritime transportation systems with similarities to Short Sea Pioneer

Feeder vessels in container shipping is not a new concept. Feeder services are extensively used to connect small container ports to the larger hub ports served by deep-sea shipping lines (Monios, 2017). Some unconventional feeder systems with similarities to the mother daughter vessel system Short Sea Pioneer is identified as inland barge navigation, lighter abroad ships and midstream operations in Hong Kong. These systems are all briefly described in this sub-chapter.

The lighter abroad ship (LASH) is a large mother vessel designed as a barge carrier. The mothership is equipped with a large gantry crane that can pick up and lower the barges into the water. The first LASH was put into service in 1969 to avoid port congestion (Leach, 2007). The mother vessels unloaded the barges at river inlets and tugboats were used to push the barges up the rivers. This way the LASH used the inland river system to get around the problem of congested ports and low infrastructure. Leach (2007) argues that the LASH worked best for trade lines with shallow waterways or extensive river systems in each end of the route.

Ship-to-ship cargo transfer is widely used in the port of Hong Kong in so-called midstream operations (Kim & Morrison, 2012). In midstream operations container ships are unloaded and loaded at sea, relatively close to land, by use of several small barges equipped with derrick cranes. After the unloading and loading is completed, tugboats are used to transport the container barges to shallow water berths where they use their own crane to (un)load containers at berth. Midstream operations are useful in Hong Kong as port expansion is limited by geography. Berths for the container barges occupy less area compared to berths for large sea-going vessels, and only require about five-meter water depth. The barges are thus only charged 1/3 of the container handling fee in the traditional Hong Kong port (Kim & Morrison, 2012). According to Kim and Morrison (2012), direct ship-to-ship cargo transfer is seldom conducted outside of the midstream operations in Hong Kong because of the need to tightly coordinate the schedules of the involved ships. Tang et al. (2013) states that the midstream operation survives in Hong Kong because of the significant fee-gap between midstream operations and container terminal operations. They state that midstream operations present a significant cost advantage, but that marine insurance companies see it as a risky practice.

Inland shipping by barge is extensively used in European rivers. The port of Rotterdam has the most extensive inland barge shipping system, where cargo is transported through the Rhine and Maas river (Konings et al., 2010). Utilizing the rivers to access the hinterland allows the cargo to travel straight past traffic congestion, which allows for a reliable shipping service (Portofrotterdam). Unlike midstream operations in Hong Kong, the cargo serviced by the inland barges passes through a land terminal when it is transferred from the large sea-going container vessels to the river barges. Intermediate storing of the containers at a land terminal reduces the need for tight coordination between sea-going vessels and inland barges. However, handling sea-going vessels and barges at the same terminals may lead to increased waiting times for the barges as the terminals prioritize sea-going vessels because they are significantly costlier to delay (Caris et al., 2011).

4.2 Modelling of Maritime Transportation Systems

Fazi et al. (2015) construct an optimization model as a decision support tool for barge transportation of import containers. The decision support model generates schedules for the inland barges from sea terminals to terminals in the hinterland. The model also determines which containers that should be transported on a barge and which containers that should be transported by truck. They conclude that quick and smart planning is necessary to improve the flexibility and the performance of hinterland barge transportation.

Caris et al. (2011) model the inland container barge transportation in the port of Antwerp by use of simulation. They evaluate different hub scenarios' impact on waiting times and capacity utilization. The background for the study is increasing waiting times for inland barges in the port of Antwerp. They emphasize the importance of reorganizing the container barge services as sea terminals prioritize sea-going vessels before inland barges which results in delays for the barge services. They find that introduction of an intermodal barge hub might lead to reduced turnaround times for the inland barge services and to increased efficiency in the sea terminals.

Cigolini et al. (2014) compare simulation against mixed integer linear programming (MILP) and spreadsheet models as decision support tools for offshore oil and gas supply chains. They evaluate a real case from an oil company related to the problem of sizing a fleet of feeder vessels for pipe-laying from an offshore field. The evaluated system consists of three types of vessels: A pipe-lay vessel that moves along the projected pipeline route from the offshore field towards land, a fleet of feeder vessels that transport pipe segments from an onshore transshipment point to the pipe-lay vessel, and a mother vessel that feeds the pipe segments from a port to the transshipment point. The problem they investigate is the number of feeder vessels needed and the location of the onshore transshipment point. They first introduce a deterministic simulation model that produce quite similar results as the spreadsheet model. This on one hand verifies that the simulation model work as intended and on the other hand proves that the spreadsheet model is well suited for the deterministic problem. However, by investigating the underlying assumptions for the deterministic model they identify two main weaknesses of the spreadsheet model: First, it cannot take into account the continuous movement of the pipe-lay vessel, and second, it does not consider the stochastic variables influencing the system, such as weather variation, crane breakdowns and canal traffic. They also state that the simulation model is better suited for scenario testing. They conclude that simulation represents a good alternative to spreadsheet and MILP models for sizing the elements of a logistics system.

Halvorsen-Weare et al. (2012) develop an optimization model for fleet composition and periodic routing of offshore supply vessels in cooperation with Statoil. The objective of the model is to minimize charter- and sailing costs. The problem consists of an onshore depot and a set of offshore installations with different requirements of vessel visits during the planning horizon. Opening hours for both the depot and the installations are considered. The problem is formulated as an integer programming model with a priori voyage generation. Opening hours are thus handled in the voyage generation, and are not mathematically modelled as constraints in the optimization model. The routes and schedule are generated to be repeated on a weekly basis. Robustness of the constructed routes and schedules are only briefly evaluated, and are handled by forcing slack in the solutions by setting an upper limit of voyage duration that is somewhat lower than the available time. The voyage based solution method is implemented as a planning tool at Statoil, who estimate cost savings of roughly USD 3 million per year. For future research, they emphasize the importance of creating robust schedules that can cope with small delays without consequences for future voyages.

Agra et al. (2013) develop a MILP model for short sea fuel oil distribution where port opening hours are considered. Unlike Halvorsen-Weare et al. (2012), they model the problem as an arc-flow model and hence have to model port opening hours as constraints to the optimization model, not in a priori generation of voyages as done by Halvorsen-Weare et al. (2012).

5. Methodology

This chapter describes the methodologies used to develop the decision support models. First, a description of the combined optimization-simulation approach is given before optimization and simulation are further described on a generic basis. At the end of the chapter, the methodology for estimating operability of the ship-to-ship cargo transfer operation is presented.

5.1 Combined Optimization-simulation approach

A combined optimization-simulation approach is used to establish the daughter vessel's speed and routes. As illustrated in Figure 3, the optimization model is used to determine the daughter vessel's minimum speed and routes, based on deterministic input data. The output from the optimization model: speed and routes, are then used as input in the simulation model together with stochastic input data, such as variations in transportation demand and cargo transfer rates. The simulation model is used to assess the performance of the daughter vessel feeder system when the system is subject to uncertainties and a dynamic nature.

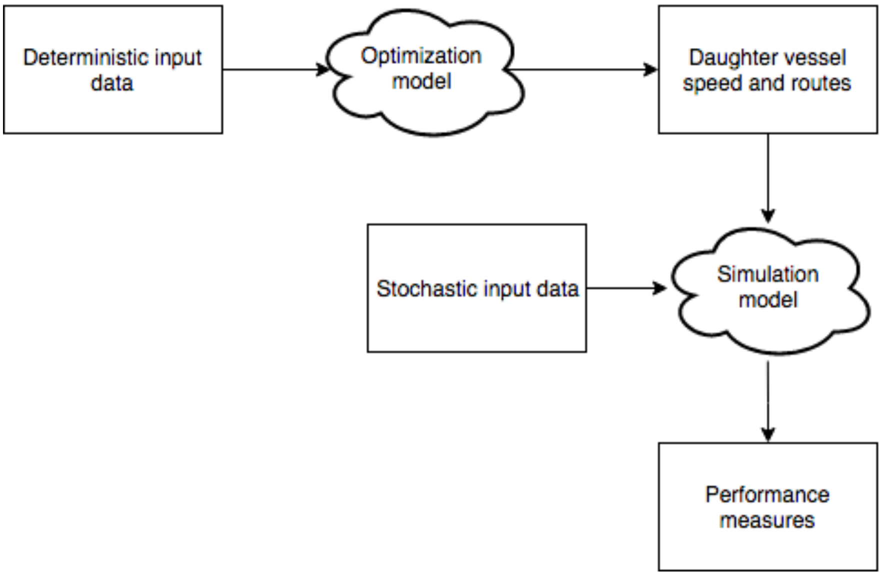


Figure 3: Combined optimization-simulation approach. Output from optimization model used as input in simulation model.

The optimization model is only run once for a given feeder case and are used to establish a base solution for vessel speed and routes. The simulation model is then used to assess performance of the feeder system. The simulation model is run several times where vessel speed is adjusted from the base solution between every run.

A combined optimization-simulation approach is chosen because optimization, or linear programming, is well suited for problems with many variables where it is hard to identify the best solution by intuition, and simulation is well-suited for problems that involve uncertainties, and for scenario testing. The optimization model is hence used to establish a preliminary solution and to delimit the feasible area, while the simulation model is used to assess performance when the system is subject to uncertainties and unforeseen events, and to adjust the parameters. In other words, the simulation model is used to evaluate and increase the robustness of the solution.

In the optimization model, it is assumed that cargo transfer between mother vessel and daughter vessel always can be carried out at sea when both vessels are present at the meeting point. On the other hand, the simulation model is run with different operabilities of the ship-to-ship cargo transfer operation. If the weather is too severe to perform cargo transfer at sea, the vessels will sail to a nearby port.

5.2 Optimization

Optimization is the use of mathematical models to find the best solutions/alternatives in decision making situations and is part of the field applied mathematics (Lundgren et al., 2010). For optimization to be used there must be something variable in the problem that can be controlled by the decision maker and the considered aspects of the problem must be quantifiable. When a problem is optimized, one determines the values of the considered decision variables that gives the best possible value for a defined objective function. The objective function can either be minimized or maximized. A set of constraints are used to consider restrictions on the decision variables in the problem. Optimization can be used for operative, tactical and strategic planning (Lundgren et al., 2010). Lundgren et al. (2010) mention transport and logistics as one of the main areas of optimization.

The optimization process involves simplification of the real-world problem, formulation of the simplified problem to an optimization model (mathematical model), application of an optimization algorithm to solve the problem, verification, and evaluation (Lundgren et al., 2010). When the real-world problem is simplified and formulated as a mathematical model there is a tradeoff between more exact description of the real-world problem and the solvability of the problem. Lundgren et al. (2010) states that additionally to the main purpose of obtaining support for decisions in the problem, the work process of formulating objective function and constraints leads to improved knowledge of the problem itself.

In this thesis, linear programming is used for the optimization process. In linear programming, the objective function to be minimized or maximized must be a linear function and the constraints to the problem must be linear inequalities and/or linear equations (Hu & Kahng, 2016). An advantage of using linear programming is that there exist efficient algorithms for solving the problems. The Simplex method, developed by George Dantzig in 1947, is one of the commonly used algorithms. The method was a breakthrough in the field of optimization, and has had great influence on science and engineering in general (Chong & çZak, 2014). A disadvantage of linear programming is that the assumption of a linear problem is not always realistic, and can be limiting on the model (Hu & Kahng, 2016). However, there exist methods for linearizing nonlinear aspects of a problem (Hillier & Lieberman, 2015).

5.3 Simulation

Simulation is a useful tool to identify bottlenecks and problems in a system before actually building or modifying the system (Carson, 2003). Simulation allows for comparison between a wide range of designs and operating contexts while it considers the dynamics of the system. It is also well-suited for studying the interaction between sub-systems or components (Carson, 2003). Banks et al. (2005) define simulation as follows:

"A simulation is the imitation of the operation of a real-world process or system over time. Whether done by hand or on a computer, simulation involves the generation of an artificial history of a system and the observation of that artificial history to draw inferences concerning the operating characteristics of the real system".

In addition to provide numerical measures for the system being studied, a good simulation model should provide insight into the system performance and give an understanding of the system behavior (Carson, 2003).

Carson (2003) identifies a range of drawbacks with simulation that should be kept in mind when analyzing a system by use of simulation: Simulation can be a time-consuming process and is therefore not applicable for all projects. Unexperienced modelers or people too fixated on the simulation software might develop models with too much details which results in project-deadline overruns. Also, necessary input data for the models might be unavailable or expensive to obtain.

5.4 Operability of ship-to-ship cargo transfer operation

Operability of the ship-to-ship cargo transfer operation is estimated for different operational criterions and different durations of the operation, based on historical weather data from the meeting point for mother and daughter vessels. The operational criterion is defined as "*maximum weather condition for execution of the marine operation*" (Larsen, 2017). By including the duration of the operation when estimating the operability, one also considers the persistence of the weather, not only the extreme value distribution (Nielsen, 2008). For the ship-to-ship cargo transfer operation it is assumed that significant wave height is the limiting parameter.

6. Models

In this chapter, all constructed models are presented and described. The optimization model is first presented without considerations of opening hours in the ports, and is later extended to include this. Thereafter the simulation model is presented, before the model for estimating operability of the ship-to-ship cargo transfer operation is presented at the end of the chapter.

6.1 Optimization model – Mixed Integer Linear Programming

The optimization model is constructed as a two-step arc-flow model where step 1 is to minimize the speed of the daughter vessel and step 2 is to minimize the distance travelled by that vessel. The speed found as output from step 1 is fixed and used as input for step 2. The input needed and the output obtained from running the model (step 1 and step 2) is presented in Table 1.

Table 1: Optimization model input and output. No direct relationship between input and output on the same line.

Input	**Output**
Distances	Minimum daughter vessel speed
Number of periods	
Arrival times of mother vessels (end time of periods)	Daughter vessel routes for each period
Required number of visits for each port during the cycle time	
Number of containers to be handled in each port per visit	
Cargo transfer rates	
Time used for mooring/unmooring with mother vessel	
Time used for mooring/unmooring in port	

The optimization model is constructed as a deterministic model where the input data are based on expectation values for the evaluated case. The model is used to determine daughter vessel speed and routes for a given cycle time. The cycle time is further divided into a number of periods, where the end of each period represents the arrival of a mother vessel. The feeder system should therefore be designed such that the daughter vessel is present at the meeting point for ship-to-ship cargo transfer at the end of each period.

Number of containers to be handled in each port at a port visit is treated as a constant. This is done because the number of visits for each port during the cycle time is determined before the optimization model is run. Number of containers to be handled between mother and daughter vessel at the transshipment meeting point is however treated as a variable because it is dependent on which ports that were visited in the previous period, and which ports that will be visited in the current period.

The different sets, parameters, and variables used in the optimization model are presented in Table 2, Table 3 and Table 4, respectively.

Table 2: Sets used in optimization model.

Set	Definition
V	Set of vessels, indexed by v
N	Set of ports, indexed by i and j
S	Subset of N, including all ports contained in N that should be visited in half of the periods or less
P	Set of periods, indexed by p

Although only use of one daughter vessel is considered, a set of vessels, v, is included in the model. This is done to linearize the model as speed is now treated as a parameter instead of a variable. The optimization model still minimizes speed of the vessel, but this is done indirectly by choosing which vessel to use instead of directly choosing vessel speed. Speed is the only parameter that varies between the different vessels. Linearizing the model by introducing a set of vessels also imply that the model is discretized for vessel speeds.

It should be noted that the origin and destination port, o and d, contained in N, are geographically identical and both correspond to the transshipment meeting point at sea. The meeting point is duplicated because the vessel both start and end here in each period. Using the duplicate node allows the time constraints to be formulated more simple. For the set of periods, P, the last period is denoted by e.

Table 3: Parameters used in optimization model.

Parameter	Definition
D_{ij}	Distance between port i and j [nautical miles]
R_i	Required number of visits for port i during cycle time (all periods)
T_p^S	Start time of period p [h]
T_p^E	End time of period p [h]
H_i	Expected number of TEUs (twenty feet equivalent units) to be handled each way at port i at each port visit
S_v	Speed of vessel v [knots]
L_i	Loading/unloading rate at port i [TEUs/hour]
T^M	Time spent for mooring and unmooring at meeting point with mother vessel [h]
T^P	Time spent for mooring and unmooring at port [h]
M_{ip}^1	Large positive number
M_{ip}^2	Large positive number

Table 4: Variables used in optimization model.

Variable	Definition
x_v	= 1 if daughter vessel v is used in the solution = 0 otherwise
u_{vijp}	= 1 if daughter vessel v sails directly from node i to node j in period p = 0 otherwise
t_{ip}	= Time when daughter vessel arrives port i in period p
a_p^L	= Number of TEUs to be loaded to daughter at meeting point in start of period p
a_p^U	= Number of TEUs to be unloaded from daughter at meeting point in start of period p

Time when daughter vessel arrives at port i in period p, t_{ip}, is not indexed with vessel, v. This is not necessary because the model makes sure that only one vessel is used in the solution. It should be noted that the variables keeping track of number of TEUs to be loaded and unloaded from/to the mother vessel, a_p^L and a_p^U could be removed and expressed through H_i and u_{vijp} in the constraints. However, they are included as helping variables to make the constraints more intuitive and easier to read.

The objective function for step 1 and step 2 are shown in (1) and (2) respectively, while the constraints for the model are expressed in (3) through (25), and are all further described below.

$$MIN\ Z = \sum_{v \in V} S_v\, x_v \tag{1}$$

$$MIN\ Z' = \sum_{v \in V} \sum_{i \in N} \sum_{j \in N} \sum_{p \in P} D_{ij} u_{vijp} \tag{2}$$

$$x_v - \sum_{j \in N} u_{vojp} \geq 0 \qquad , v \in V, p \in P \tag{3}$$

$$\sum_{v \in V} x_v = 1 \tag{4}$$

$$\sum_{v \in V} \sum_{j \in N} \sum_{p \in P} u_{vijp} = R_i \qquad , i \in N \setminus \{o, d\} \tag{5}$$

$$\sum_{i\in N} u_{vijp} - \sum_{i\in N} u_{vjip} = 0 \qquad , v \in V, j \in N \setminus \{o, d\}, p \in P \tag{6}$$

$$\sum_{v\in V}\sum_{j\in N} u_{vojp} = 1 \qquad , p \in P \tag{7}$$

$$\sum_{v\in V}\sum_{i\in N} u_{viop} = 0 \qquad , p \in P \tag{8}$$

$$\sum_{v\in V}\sum_{i\in N} u_{vidp} = 1 \qquad , p \in P \tag{9}$$

$$\sum_{v\in V}\sum_{j\in N} u_{vdjp} = 0 \qquad , p \in P \tag{10}$$

$$a_p^L = \sum_{v\in V}\sum_{i\in N\setminus 1}\sum_{j\in N} H_i\, u_{vijp} \qquad , p \in P \tag{11}$$

$$a_p^U = \ a_{p-1}^L \qquad , p \in P \setminus 1 \tag{12}$$

$$a_1^U = a_4^L \tag{13}$$

$$t_{1p} = \ T_p^S \qquad , p \in P \tag{14}$$

$$t_{jp} - t_{ip} - \frac{D_{ij}}{S_v} - T^P - \frac{2H_i}{L_i} + M_p^1\left(1 - u_{vijp}\right) \geq 0 \,, v \in V, i \in N \setminus \mathrm{o}, j \in N \setminus \mathrm{o}, p \in P \tag{15}$$

$$t_{jp} - t_{1p} - \frac{\mathrm{D}_{1j}}{S_v} - T^M - \frac{(a_p^U + a_p^L)}{L_1} + M_p^2\left(1 - u_{v1jp}\right) \geq 0 \ \ , v \in V, j \in N \setminus \mathrm{o}, p \in P \tag{16}$$

$$t_{ip} \geq T_p^S \quad \ i \in N \,, p \in P \tag{17}$$

$$t_{ip} \leq T_p^E \quad i \in N\,, p \in P \tag{18}$$

$$\sum_{v \in V}\sum_{j \in N} u_{vi'jp} + \sum_{v \in V}\sum_{j \in N} u_{vi'jp+1} \leq 1, \qquad i' \in \mathrm{S}, p \in P \setminus \mathrm{e} \tag{19}$$

$$\sum_{v \in V}\sum_{j \in N} u_{vi'j1} + \sum_{v \in V}\sum_{j \in N} u_{vi'j\mathrm{e}} \leq 1, \qquad i' \in \mathrm{S} \tag{20}$$

$$x_v \in \{0,1\} \quad , v \in V \tag{21}$$

$$u_{ijp} \in \{0,1\} \quad , i,j \in N, p \in P \tag{22}$$

$$t_{ip} \geq 0 \quad , i,j \in N, p \in P \tag{23}$$

$$a_p^L \geq 0 \text{ and integer} \qquad , p \in P \tag{24}$$

$$a_p^U \geq 0 \text{ and integer} \qquad , p \in P \tag{25}$$

The model is first solved with the objective function for step 1 (1) that choose the vessel with the lowest possible speed. Thereafter, the model is solved with the objective function for step 2 (2) that minimize the total distance sailed by the vessel during the entire planning horizon. When step 2 is solved, the vessel found as solution from step 1 is forced to be used. This is done by setting $x_v = 1$ for that specific vessel before solving step 2 of the model.

Constraint (3) makes sure that the decision variable x_v takes the value 1 if vessel v is used in the solution, while constraint (4) makes sure that exactly one vessel is used. Constraint (4) is only required when solving step 2 of the model as the objective function from step 1 makes sure that as few vessels as possible are used. (5) assures that each port is visited it's required number of times during the cycle time. (6) is a flow-conservation constraint. It makes sure that if the vessel sails to a port in a period, it must also leave that port during the same period. This constraint is not defined for the origin and destination node. Constraint (7) and (9) make sure that the vessel start each period from the origin node and end each period at the destination node, while constraint (8) and (10) assures that the vessel cannot sail to the origin node or from the destination node.

Constraints (11) through (13) keep track of the number of TEUs to be loaded and unloaded at meeting point for each period, a_p^L and a_p^U. The number of TEUs to be loaded from the mother vessel in a period is determined by which ports it will visit in that period and the number of TEUs to be delivered in those ports. This is expressed by (11). Because it is assumed perfect balance between ingoing and outgoing containers for each port, the number of containers to be unloaded to the mother vessel is equal to the number of containers delivered in the previous period. This is modelled by (12), which is defined for all except the first period. Because of the cyclicality of the routes, the number of containers to be unloaded to mother vessel in the first period is equal to the number of containers picked up in the last period. This is modelled by (13).

Constraints (14) to (18) are all time-related. (14) makes sure that the daughter vessel starts operation at the transshipment meeting point at the beginning of each period. This is valid because the solution is not feasible if the vessel does not make it back to the meeting point before the end of a period. (15) makes sure that the time variable is updated at each port visit based on the time of arrival in the previous port, the time used for mooring/unmooring and loading/unloading in that port, and the time spent sailing between that port and the current port. This constraint should only be restrictive if the port in question is visited in the current period. This is handled by M_{ip}^1. (16) is comparable to (12) but is only valid for the port that is visited directly after the meeting with the mother vessel. This is handled by M_p^2. Arrival time in the port visited directly after the transshipment meeting point is determined by the start time of the current period, the time used for unloading to- and loading from the mother vessel, and the time used for sailing from the meeting point, o, to the port in question. Constraint (17) and (18) makes sure that a port can only be visited in a period if the current time is equal to or more than the start time of that period and equal to or less than the end time of that period. Time constraints (14) to (18) also prevents sub-tours to be part of the solution as a port cannot be visited at different times during the same period.

Constraints (19) and (20) makes sure that a port that should be visited in half of the periods or less are not visited two periods in a row. This ensures spread in the port visits.

(21) and (22) are binary constraints while (23) is a non-negativity constraint. (24) and (25) are integer constraints. These integer constraints can be removed because a_p^L and a_p^U are forced to always take the value of a positive integer by (12) to (14).

M_{ip}^1 and M_{ip}^2 are given the value T_p^E when solving the model. This is reasonable because a port can never be visited in a period if the time exceeds the end of that period. It also gives relatively tight constraints. However, by decreasing the value of M as expressed in (26) and (27), one will tighten the constraints further. This could be implemented if the solution time of the model gets high

$$M_{ip}^1 = \max\left\{ t_{ip} + \frac{D_{ij}}{S_v} + T^P + \frac{2H_i}{L_i} \right\} \quad , v \in V, i \in N \setminus \text{o}, j \in N \setminus \text{o}, p \in P \tag{26}$$

$$M_{ip}^2 = max\left\{ t_{\text{o}p} + \frac{D_{\text{o}j}}{S_v} + T^M + \frac{\left(a_p^U + a_p^L\right)}{L_\text{o}} \right\} \quad , v \in V, j \in N \setminus \text{o}, p \in P \tag{27}$$

When the model is implemented and solved in a software, additional constraints (28) and (29) can be added. These constraints are not necessary to solve the model but are convenient for reading the results as they force the arrival time in the different ports to their exact values. As they are not necessary for the model they should be removed if the problem is large and the number of constraints gets too high.

$$t_{jp} - t_{ip} - \frac{D_{ij}}{S_v} - T^P - \frac{2H_i}{L_i} + M_p^1\left(1 - u_{vijp}\right) \leq 0 \, , v \in V, i \in N \setminus o, j \in N \setminus o, p \in P \tag{28}$$

$$t_{jp} - t_{op} - \frac{D_{\text{o}j}}{S_v} - T^M - \frac{(a_p^U + a_p^L)}{L_\text{o}} - M_p^2\left(1 - u_{v\text{o}j\text{p}}\right) \leq 0 \quad , v \in V, j \in N \setminus o, p \in P \tag{29}$$

6.1.1 Extended model to include opening hours in ports

The model is extended to consider port opening hours. The opening hours are treated as hard time windows, meaning that a vessel arriving outside the opening hours can under no circumstances start port operation until the port opens. It is assumed that when the vessel starts port operations, it is allowed to finish loading/unloading even though the operation exceeds the opening hours of the port.

To include port opening hours in the model, two additional parameters and one additional variable are introduced as presented in Table 5 and Table 6.

Table 5:Additional parameters for optimization model to include opening hours in ports.

Symbols	**Definitions**
T_i^O	Opening hour for port i [h]
T_i^C	Closing hour for port i [h]

Table 6: Additional variable for optimization model to include opening hours in ports.

Symbols	**Definitions**
d_{ip}	= Day that port i is visited in period p

The variable d_{ip} is introduced to keep track of which day a port is visited in each period. This variable is introduced to model cyclicality of port opening hours, as a port is assumed to open and close at the same time every day. All constraints used to model the port opening hours are expressed in (30) to (34).

$$d_{ip} \leq \frac{t_{ip}}{24} \qquad , i \in N \setminus \{o, d\}, p \in P \tag{30}$$

$$d_{ip} \geq \frac{t_{ip}}{24} - 1 \qquad , i \in N \setminus \{o, d\}, p \in P \tag{31}$$

$$t_{ip} - 24 d_{ip} \geq T_i^O \qquad , i \in N \setminus \{o, d\}, p \in P \tag{32}$$

$$t_{ip} - 24 d_{ip} \leq T_i^C \qquad , i \in N \setminus \{o, d\}, p \in P \tag{33}$$

$$d_{ip} \geq 0 \; and \; integer \quad , i \in \mathrm{N}, p \in P \tag{34}$$

Constraint (30) and (31) are used to connect the day that port i is visited, d_{ip}, to the time that port i is visited, t_{ip}, and are necessary for the modelling of (32) and (33). Constraint (32) and (33) assures that a port can only be visited after the port is opened for the day and before the port is closed for the day.

When the model is solved with opening hours in ports, constraint (28) and (29) must be removed as the vessel sometimes has to wait outside a port until it opens before it enters. If (28) and (29) are not removed, these solutions will be unfeasible, and the best solutions might get lost.

It should be noted that the model can still be used for problems with 24/7 port service after the additional constraints for opening hours are implemented by setting $T_i^O = 0$ and $T_i^C = 24$.

However, it could be convenient to remove the constraints when they are not needed to reduce the number of constraints in the problem.

6.2 Simulation model

The simulation model is constructed in MATLAB SimEvents, which is a discrete-event simulation engine including a component library (MathWorks, 2017a). SimEvents provide a graphical modelling environment that allows for more intuitive model-building compared to script models.

In this chapter, only the main components and functions of the simulation model are described. For detailed description of the model it is referred to Appendix C where all coding and configuration of the different system blocks and functions are presented.

The input used to run the simulation model and the output obtained from running it is presented in Table 7.

Table 7: Simulation model input and output. No direct relationship between input and output on the same line.

Input	Output
Daughter vessel speed	Share of meetings that the daughter vessel arrives on time
Cycle time	
Number of periods in cycle	Maximum, minimum and average time spent waiting for the mother vessel
Daughter vessel route for each period	
Sailing distances	
Mother vessel arrival time for each period	
Operability of ship-to-ship cargo transfer operation	
Cargo transfer rates for the different ports and for ship-to-ship cargo transfer operation	
Transportation demand for each port	
Opening hour for ports	
Closing hour for ports	

All input parameters except from speed and operability of the ship-to-ship cargo transfer operation are configure in Excel. Speed and operability of ship-to-ship cargo transfer operation are configured in the script used to run the model.

Two Excel documents are configured, one document containing the distance matrix, and the other document containing the remaining input parameters. These documents can be found in the end of Appendix D.

The main components and architecture of the constructed simulation model, excluding all Simulink functions and global variables, are shown in Figure 4, and are further described in the following sub-chapters with reference to the figure.

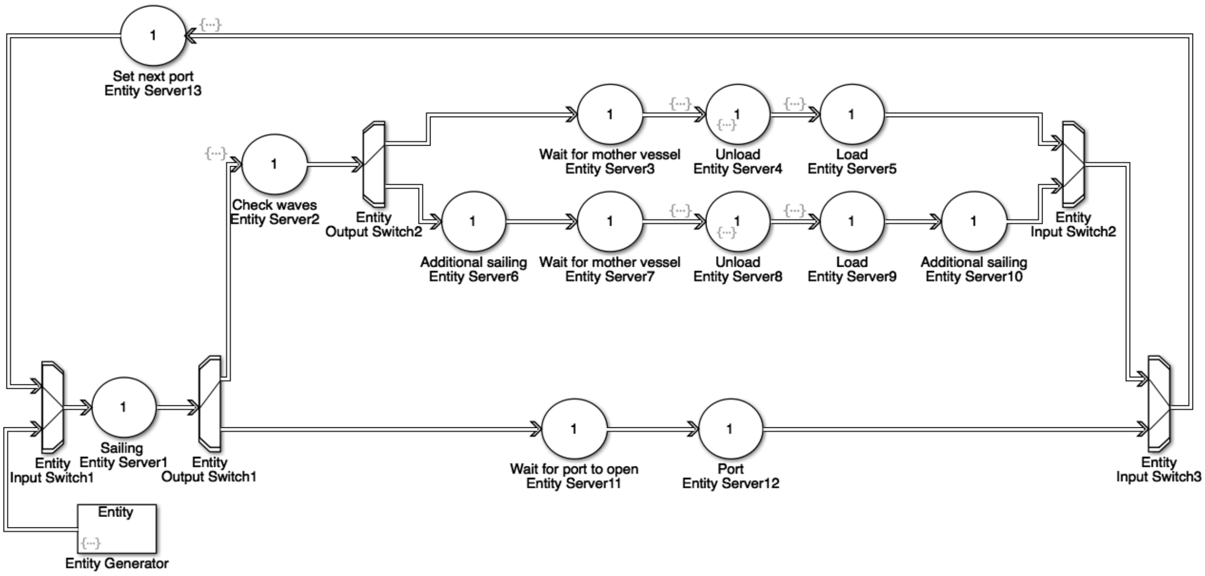


Figure 4: Simulation model as constructed in SimEvents. Excluding all Simulink functions and global variables.

To make the model generic and applicable for different problem sizes, the different ports are not physically separated in the model. All port operations find place in entity server 12, while all sailing finds place in entity server 13. By configuring the model this way, the problem can be extended to an arbitrary number of ports without having to change the physical configuration of the model. Ship-to-ship cargo transfer operation is however separated from port operations because of the vastly different nature of these operations. Separating ship-to-ship cargo transfer operations from port operations also simplifies the process of logging arrival times at the meeting point. The meeting point for ship-to-ship cargo transfer should always be treated as port 1 when using the model.

The daughter vessel is treated as an entity in the model, and is guided through the different components based on the pre-generated routes from the optimization model. A set of attributes is assigned to the vessel (entity) describing different characteristics of the vessel. All attributes are briefly explained in Table 8.

Table 8: Attributes assigned to the vessel (entity) in the simulation model.

Attribute name	Attribute description
Speed	Speed of vessel [knots]
From	Previous visited port
To	Next port of visit
PortCall	Port call number in period
Period	Period number
Cycle	Cycle number
Loaded	Number of TEUs on vessel
Wave	Used to guide daughter vessel to meeting port if waves are too severe
Switch	Used to guide daughter vessel either to a port or to the transshipment meeting point based destination (attribute To)

All attributes follow the vessel throughout the model and can hence be accessed and modified in all system components that the vessel passes through.

6.2.1 Routing the vessel in the model

The routes found as output from the optimization model are used as input for the simulation model and are configured in the Excel document "Fleet" from Appendix D. At the start of the simulation, when the vessel is deployed in the model, attribute From and attribute To are both assigned the value 1, which corresponds to the transshipment meeting point. When attribute To is equal to attribute From, the distance to be sailed is 0. The vessel hence starts from the transshipment meeting point.

The vessel is routed by always keeping track of where the vessel is sailing from and to, which period the vessel is sailing in, and how many port calls it has completed in the current period. After the vessel completes service in a port or transshipment at the meeting point, it enters entity server 13 where it reads the next destination from the input Excel sheet based on which period it is sailing in (attribute Period) and how many port calls it has completed in the current period (attribute PortCall). After the next port of visit is determined, the vessel enters entity server 1 where the distance to be sailed is obtained by accessing the element in the distance matrix corresponding to attribute From and To.

After sailing is completed the vessel enters Entity Output Switch1 where the vessel either is sent to a port or to the transshipment meeting point, based on attribute Switch. If the next destination is a port, attribute switch is set to 2 and the vessel is sent to entity server 11. In entity server 11, the current time is checked against opening hours for the port. If the port is open, the vessel is immediately forwarded to entity server 12 without using any time in entity server 11. If the port is closed, the time until it opens is calculated, and the vessel stays in entity server 11 until the port opens. In entity server 12, loading and unloading in the respective port is carried out.

If the next destination is the transshipment meeting point when the vessel enters Entity Output Switch 1, attribute Switch is set to 1. The vessel is then sent to entity server 2 where it checks if the ship-to-ship cargo transfer can be carried out at sea. If the weather is unfavorable the vessel is routed to Entity Server6 where it sails an additional distance to reach a sheltered port where it waits for the mother vessel. If the weather condition is within the operational limit of the ship-to-ship cargo transfer at sea, the vessel is sent to entity server 3 where it waits for the arrival of a mother vessel. For ship-to-ship cargo transfer at sea, unloading and loading finds place in entity server 4 and 5, while for ship-to-ship cargo transfer sheltered in port it finds place in entity server 8 and 9.

6.2.2 Cycling on routes from Excel

Daughter vessel routes for the different periods is separated by different columns in the Excel input document. When routing the ships in the model, attribute Period is used to access the column corresponding to the route for the current period. When the daughter vessel starts unloading to the mother vessel (on arrival in entity server 4 or 8) entity Period is updated. If the current period is less than the total number of periods in the cycle, attribute Period is increased by 1. Otherwise, if the current period is equal to the total number of periods in the cycle, attribute Period is set to 1. This way the daughter vessel cycle over the generated routes for the entire simulation time.

6.2.3 Global variables and Simulink functions

Each MATLAB function in the different components in the model have their own local variables, that are not available between different components. Global variables are therefore used for variables that are accessed in different components of the model: time and cargo on vessel including destination information. By using global variables, these variables can be accessed and used between different functions. If the variable is changed in any of the functions, this change is visible in all functions where it is declared as global (MathWorks, 2017b).

Global variables are not supported in the components from the SimEvents library. Simulink functions are therefore implemented in the model to allow for use of global variables. Input to the different Simulink functions are defined in the system block where they are called upon, and output from the functions are returned to that system block. A screenshot of the model including function connectors to the Simulink functions can be found in Appendix C.

6.2.4 Improving runtime of the model

Parameters from the two input Excel documents are frequently accessed from different Entity Servers and Simulink functions to determine routing, distances, and waiting times. When using the xlsread function in SimEvents, the Microsoft Excel file is retrieved from where it is stored on the device. To avoid this time-consuming process to be carried out every time parameters from an Excel sheet is used, the Excel sheets are stored in a persistent matrix the first time they are accessed. Persistent variables allows for preserving the state of the variables between function invocations (MathWorks, 2017c). When the vessel arrives in this component later in the simulation, the values from the Excel document is hence available from the persistent variable. The xlsread function does therefore not need to be used again. Modelling it this way makes sure that the xlsread function is maximum used one time per Excel document for a given component. The coding used to store an Excel sheet in a persistent variable is shown for entity server 1 in Appendix C.

6.2.5 Logging output data

As the simulation model is used to evaluate the robustness of the routes generated in the optimization model, arrival times at the meeting point for ship-to-ship cargo transfer with the mother vessel is an important performance measure. The time spent waiting for the mother vessel in entity server 3 or 7 is therefore sent to MATLAB workspace for every period. By sending the data to MATLAB workspace, it is available after the simulation is completed, and can be post-processed within the MATLAB environment.

6.2.6 Running simulation model from script with different input parameters

The simulation model is run several times with adjustments of input parameters between each model run. To do this in an efficient way, a MATLAB script is made for running the model. The script can be found in Appendix D together with the input Excel documents. By running the model from the MATLAB script, one can run the model for a range of input parameters without having to change the parameters manually in the SimEvents blocks, or in the Excel document, between each run. Input parameters defined in the script are accessed in the simulation model by use of constant blocks and Simulink functions.

6.3 Operability of ship-to-ship cargo transfer operation

When significant wave height (Hs) is the limiting parameter for carrying out the ship-to-ship cargo transfer operation, the availability of the operation can be calculated as follows (Larsen, 2017):

$$Operation\ availability = F_{Hs}(OP_{WF}) \cdot P(\tau_c(OP_{WF}) > T_R) \tag{35}$$

where $F_{Hs}(OP_{WF})$ is the probability that Hs is lower than the operational limit, OP$_{WF}$, and $P(\tau_c(OP_{WF}) > T_R)$ is the probability that the calm period is longer than the duration of the operation.

$F_{Hs}(OP_{WF})$ is found from the cumulative probability distribution of wave height, which in turn is found by fitting a Weibull distribution to the empirical cumulative probability distribution for the historical wave data (Larsen, 2017).

$P(\tau_c(OP_{WF}) > T_R)$ is found from the cumulative probability distribution of length of calms. This is calculated for each evaluated operational criterion by analyzing all calm periods in the hindcast data for the operational criterion in question. Figure 5 shows how storms and calms are defined when operational criterion is Hs=0.5 meters for a small section of a hindcast data. The horizontal solid lines illustrate calms while the horizontal dotted lines illustrate storms.

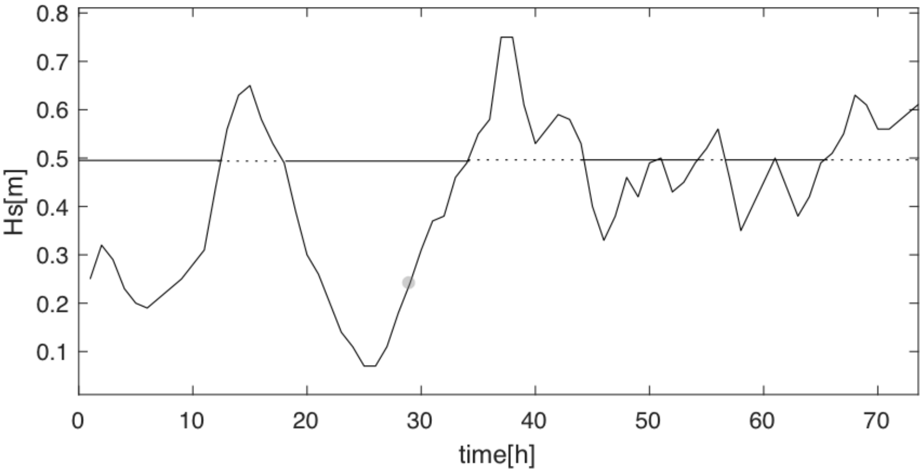


Figure 5: Definition of storms and calms. Calms illustrated by solid horizontal lines.

The cumulative probability distribution of length of calms for a specific operational criterion is found by first calculating the length of all calm periods in the hindcast data and then fitting a Weibull distribution to the empirical cumulative distribution for length of the calms. This process is repeated for each evaluated operational criterion.

After the cumulative probability distribution of wave height and the cumulative probability distributions of length of calms for all evaluated operational criterions are estimated, the expected availability of the operation is calculated by (35).

A MATLAB script is constructed to implement the above described procedure in a time-efficient manner. The script can be found in Appendix F. The script is made in a generic way and can be used for different time series for different durations and operational criterions. It should be noted that the script is constructed for time-series with data points collected for every hour. For time-series with different sampling-intervals, small adjustments must be made to the script.

7. Computational study

In this chapter the models are applied on a constructed feeder problem for the Trondheim Fjord to demonstrate potential application areas of the models and to verify that the models work as intended.

First, an overview of the considered problem is given before input data for the models are presented. Thereafter, historical wave data is analyzed for the transshipment meeting point, before the optimization model and simulation model are applied on a set of different feeder cases for the area. All routes generated from the optimization model are evaluated by use of the simulation model, both with deterministic and stochastic input data to evaluate the robustness of the different solutions. Results from each case are only briefly described where they are presented, and are further discussed in a comparative way at the end of the chapter.

Four different cases are considered: A1, A2, B1, and B2. For A1 and A2 the ports are open 24/7, while for B1 and B2 the port opening hours is limited to 07.00 to 15.00. What separates A1 from A2 and B1 from B2 is the transportation demands used for constructing routes in the optimization model. In A1 and B1, expectation values are used for transportation demands, while for A2 and B2, the 90th percentile of transportation demands are used. When the routes are tested in the simulation model, the same values for transportation demands are used for all cases.

All optimizations are run in FICO Xpress-IVE Version 1.24.15. Xpress is not supported for iOS, and all optimization runs are therefore performed on remote desktop. The generated code used to implement the model in IVE Xpress and the input file for the model can be found in Appendix B. When the speed of the daughter vessel is minimized, step 1 of the optimization model, the evaluated speeds are discretized with a step-length of 0.5 knots. The step-length is set relatively rough due to the high underlying uncertainty in the problem.

All simulations are performed in MATLAB R2016b, SimEvents, on a MacBook Pro with 2.4 Ghz Intel Core i5 processor and 4GB 1333 MHz DDR3 RAM. When results from the simulation model are analyzed, the output from the first cycle are excluded to remove start-up effects from the simulations. All simulations are run for 200 cycles.

7.1 Problem overview and context

The evaluated feeder problem is illustrated in Figure 6 ,where the red circles represent ports that should be serviced by the daughter vessel, while the orange circle (Brekstad) represents the transshipment meeting point where the daughter vessel will meet the mother vessel to perform cargo transfer at sea. The red star illustrates the sheltered port, Sandstad, where cargo transfer between mother and daughter vessel will find place if the weather conditions are too severe to carry out this operation at sea.

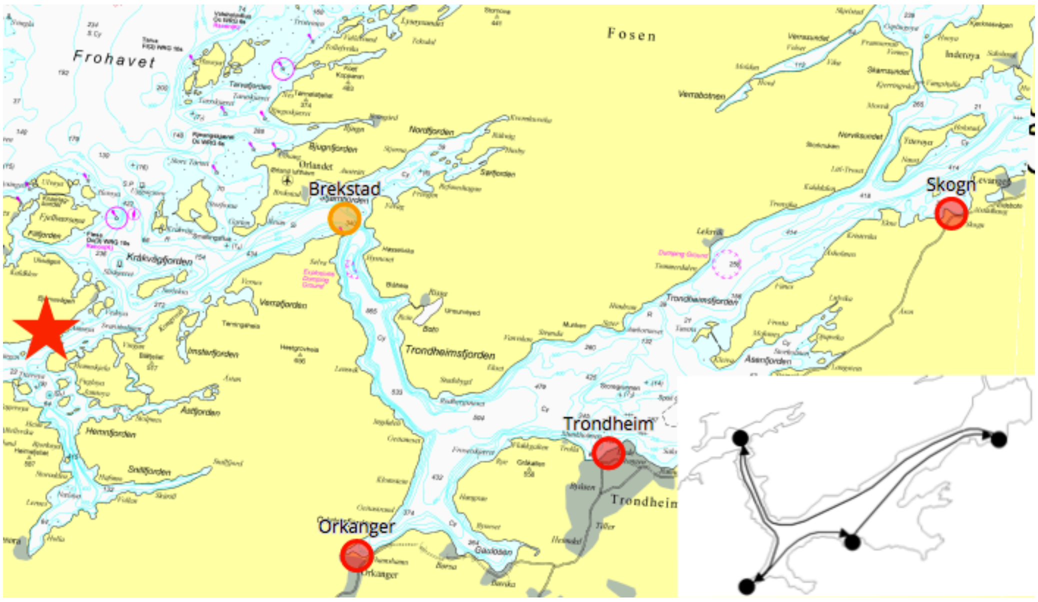


Figure 6: Illustration of feeder problem. Red circles represent ports that should be serviced by the daughter vessel, while the orange circle represent the meeting point for ship-to-ship cargo transfer. Red star illustrates sheltered port.

The ports that are included and should be serviced by the daughter vessel are Orkanger, Trondheim and Skogn. An example of a feeder route constructed by the optimization model is illustrated in the lower right corner to provide the reader with a mapping from the illustrated problem to how the ports and meeting point are presented in the results from the optimization model. In the example route the daughter vessel travels from the transshipment meeting point to Orkanger, from Orkanger to Trondheim, and from Trondheim to Skogn, before it returns to the transshipment meeting point to meet up with the mother vessel.

The Trondheim Fjord is a potentially suited area for daughter vessel feeder services because the ports in the area are located relatively far inside the fjord, which means that vessels serving these ports use considerable time sailing in the fjord. There are also a lot of industry in the area, and some of the ports are connected to rail road. The Trondheim Fjord is also the world's first test area for autonomous ships, which is relevant for short distance feeder services (Kongsberg, 2017).

Orkanger is chosen as one of the ports to include because it is the main container port in Trondheim Port Authority today. Trondheim is chosen because there is some container activity in the port today and because it is connected to the railway. Skogn is included because it has large unused capacity, is connected to the railway through own sidetrack, and has good connections to European route E6 (TrondheimPortAuthority, 2016).

Historical container moves for Trondheim Port Authority are obtained from Statistics Norway's Maritime Transport Statistics (SSB, 2016) for Q1 2003 to Q4 2016. Figure 7 shows the number of TEUs unloaded from and loaded onto ships in Trondheim Port Authority on a yearly basis in this period.

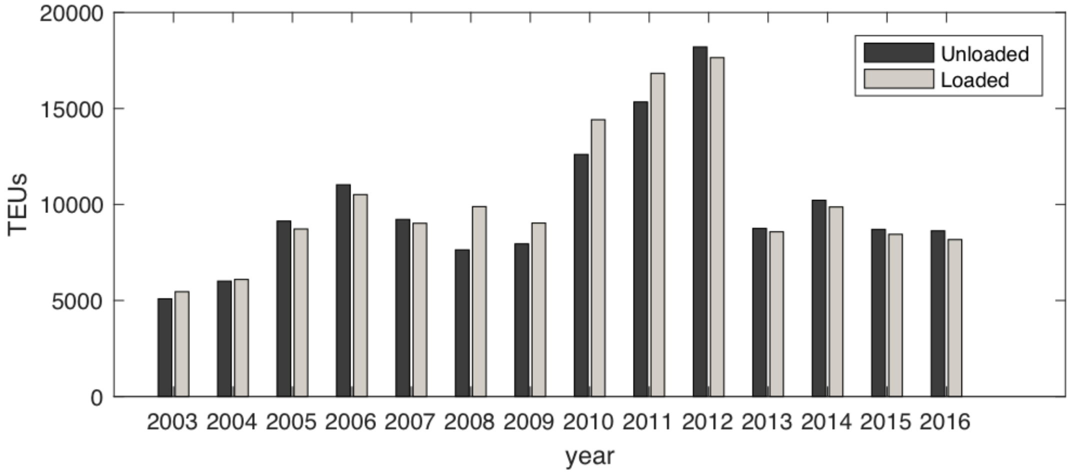


Figure 7: Historical container moves in Trondheim Port Authority 2003-2016.

It is seen that the balance between unloaded and loaded containers is relatively good. This does however not necessarily imply that there is a good balance between ingoing and outgoing commodities as containers are typically shipped empty from import ports to export ports (Huang et al., 2015). However, no distinguish is made between empty and loaded containers in this study.

Statistics from each of the ports grouped in Trondheim Port Authority are not available from Statistics Norway to prevent competitive information to be presented for the pubic. From personal conversation with a representative from Trondheim Port Authority it is clarified that the significant majority of container moves find place in Orkanger.

On realization of Short Sea Pioneer, with regular container vessel arrivals in Trondheim and Skogn, in addition to Orkanger, it is reasonable to assume that the relative share of container moves in Trondheim and Skogn will increase. For the evaluated cases, it is assumed that 80%

of all container moves will be in Orkanger, 5% in Trondheim, and 15% in Skogn. It is further assumed that the total container traffic in Trondheim Port Authority will increase by 10% compared to the average activity level in the period 2003 to 2016 because of more frequent departures and political pressure.

For the evaluated problem, it is assumed perfect balance between ingoing and outgoing containers in all ports in all periods. That is, every time a port is visited the number of containers unloaded is equal to the number of containers loaded. To generate container transportation demands, the number of ingoing and outgoing containers are therefore evaluated together. The transportation demand in one period is further assumed to be independent of the transportation demand in the previous period.

7.2 Data

In this sub-chapter the data used as input for the computational study is presented.

7.2.1 Historical wave data

Historical wave data are provided from the Norwegian Meteorological Institute from a forecast model for the Trondheim Fjord. The coordinates for the evaluated meeting point is [63.670, 9.745], which is just outside of Brekstad. The evaluated time series of significant wave height is presented in Figure 8.

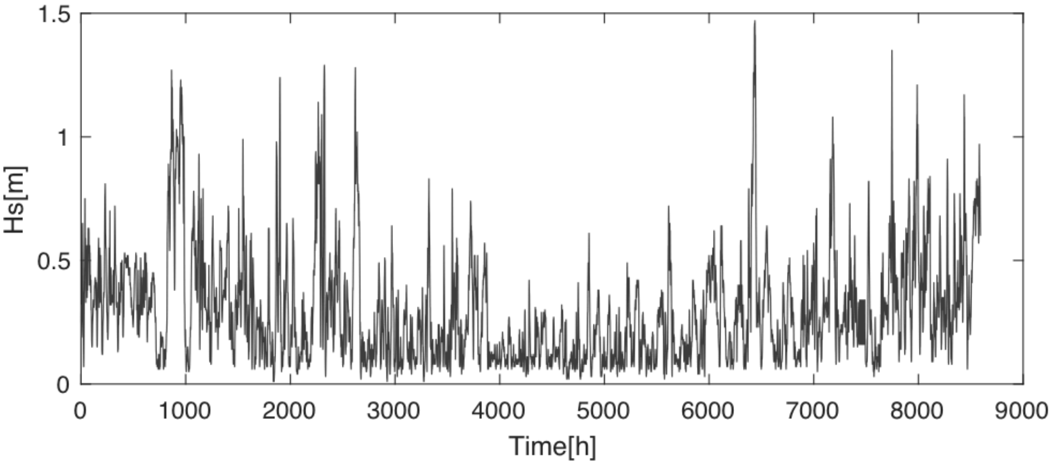


Figure 8: Forecasted significant wave height at Brekstad from January 2015 to January 2016.

Significant wave height is given for every hour for a full year. Linear interpolation is used between the data points to construct the continuous time series.

7.2.2 Cycle time and mother vessel arrivals

The evaluated cycle time is 14 days which again is divided into four periods. The routes constructed for the four periods will hence repeat themselves after every 14 days. A mother vessel arrives in the area every Monday at 06.00 and every Thursday at 18.00. It is assumed that the mother vessel always arrives on time.

7.2.3 Sailing distances

The sailing distances between the ports and meeting point are presented in the distance matrix in Table 9, where all distances are shown in nautical miles.

Table 9: Distance matrix [nautical miles]

	Brekstad	**Orkanger**	**Trondheim**	**Skogn**
Brekstad	0.0	24.2	25.4	49.7
Orkanger	24.2	0.0	19.0	44.4
Trondheim	25.3	19.0	0.0	29.4
Skogn	49.7	44.4	29.4	0.0

The distances between the different ports and transshipment meeting point are found by using the Norwegian Coastal Administration’s map tool (NCA, 2017). The distance between the transshipment meeting point outside of Brekstad and the alternative sheltered meeting point in port (Sandstad) is 20 nautical miles.

7.2.4 Cargo transfer rates and mooring times

Cargo transfer rates for the different ports and for the ship-to-ship cargo transfer operation are assumed to be triangular distributed. The lower limit, mode, upper limit, and mean of the distribution for the different ports are presented in Table 10.

Table 10: Triangular distribution for the different ports and the offshore meeting point. All values in TEUs/hour.

Port	**Lower limit**	**Peak**	**Upper limit**	**Mean**
Transshipment point	20	40	40	33
Orkanger	10	15	20	15
Trondheim	8	10	12	10
Skogn	8	10	12	10

Cargo transfer between mother and daughter vessel at the transshipment meeting point have the highest rate because containers will be handled in racks between the vessels. The upper limit is set equal to the mode, which is the desired cargo transfer rate. The lower limit is significantly lower than the upper limit and mode because the ship-to-ship container transfer technology is not yet tested and is therefore associated with high uncertainty. The cargo transfer rate in Orkanger is higher than in Trondheim and Skogn because it is the main container terminal in the Trondheim Fjord, and is therefore assumed to have more sophisticated facilities. It is assumed that containers are loaded in and out of the racks on the daughter vessel in port. Cargo transfer rates in ports are therefore lower than for cargo transfer between the vessels.

The total time used for mooring and unmooring in port and at the transshipment meeting point for each visit is 1 hour and 2 hours, respectively.

7.2.5 Container volumes and required frequency of visits for the ports

Expected transportation demand for the case is calculated as the average transportation demand from 2003-2016 plus a 10% increase. The demand is assumed to vary by ± 10% from the average value, following a uniform distribution. The MATLAB script used to calculate demands can be found in Appendix E together with the structure of the input data file from Statistics Norway.

The expected outgoing transportation demand, required number of visits for one cycle, and the expected number of containers to be loaded per visit for each port is presented in Table 11.

Table 11: Outgoing transportation demand and required number of visits for each port.

Port	Expected outgoing transportation demand (TEUs/period)	Required number of visits per 14 days	Expected number of containers to be loaded per visit	Number of containers to be loaded per visit, 90th percentile
Orkanger	92	4	92	99
Trondheim	6	2	12	13
Skogn	17	3	23	25

The required number of visits for each port during the cycle time of 14 days is determined based on the expected build-up of containers in each port per period. It is also assumed that each port must be visited at least once every week to make the service attractive for customers.

7.2.6 Daughter vessel capacity

The daughter vessel is assumed to always have enough capacity to accommodate the transportation demand. For the evaluated expectation value of transportation demand and the variation of ± 10%, this corresponds to a capacity of about 140 TEU.

7.3 Expected operability of ship-to-ship cargo transfer operation

To estimate operability of the ship-to-ship cargo transfer operation, the time series presented in Figure 8 is loaded into the MATLAB script described in Chapter 6.3 and presented in Appendix F. The operational limits that are evaluated are Hs=0.5m, 0.75m, 1m, 1.25m, and 1.5m. As seen from Figure 8, the forecasted significant wave height for the evaluated period never exceeds 1.5m. It is therefore meaningless to analyze operational limits higher than this.

The estimated availability of the ship-to-ship cargo transfer operation as function of operational limit (OPLim) and duration of the operation is presented in Figure 9.

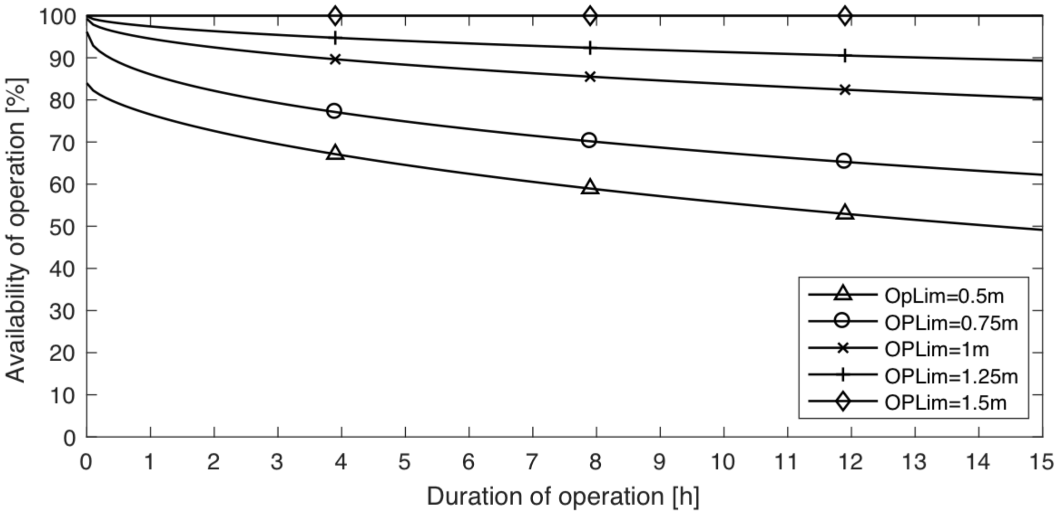


Figure 9: Availability of ship-to-ship cargo transfer operation as function of operational limit (OPLim) and duration of the operation.

It is evident that the expected availability/operability of the operation decrease with increasing duration of the operation. This decreasing effect is more significant for operational limits of 0.5 meters and 0.75 meters compared to the higher levels of operational limits. It is also seen that the increase in availability by increasing the operational limit by 0.25 meters is highest when the operational limit is increased from 0.75 meters to 1 meter.

Considering the large drop in operability for operational criterions below 1 meter, it is assumed that the ship-to-ship cargo transfer system is designed for at least 1 meter significant wave height. In further analysis, operability of 80%, 90% and 100% is therefore used.

7.4 Case A1 - Designing routes for expectation values and 24/7 port service

In this case, routes and minimum speed are determined based on 24/7 opening hours in the ports and expectation values (mean) for cargo transfer rates and number of containers to be loaded/unloaded in each port.

7.4.1 Routes generated from optimization model

The minimum speed found from step 1 of the optimization model is 2.5 knots. The preliminary constructed routes found from step 1 is described in Table 12. The meeting point for ship-to-ship cargo transfer is hereby simply referred to as Meeting.

Table 12: Description of preliminary routes from step 1 of optimization model. Expectation values of transportation demand and 24/7 port service.

Period #	Route description
1	Meeting – Orkanger – Trondheim – Meeting
2	Meeting – Skogn – Orkanger - Meeting
3	Meeting – Skogn – Trondheim – Orkanger – Meeting
4	Meeting – Orkanger – Skogn - Meeting

The total sailing distance for one cycle (all four periods) based on the routes from step 1 of the optimization model is 427 nautical miles (NM).

When step two of the model is solved, the speed of the vessel is locked to 2.5 knots and the total distance sailed over the cycle is minimized. The new routes found as output from step 2 is described in Table 13 and illustrated in Figure 10. For this case the time spent waiting for the mother vessel at Meeting is also calculated to allow for verification of the simulation model. The time spent waiting for the mother vessel at Meeting in each period is calculated from the arrival time at the meeting point and the end of the period/arrival of a mother vessel.

Table 13: Description of routes from step 2 of optimization model. Expectation values of transportation demand and 24/7 port service.

Period #	Route description	Arrival time at Meeting [h]	Time spent waiting for mother vessel [h]
1	Meeting – Orkanger – Trondheim – Skogn – Meeting	79.8	4.2
2	Meeting – Skogn – Orkanger – Meeting	159.5	8.5
3	Meeting – Orkanger – Trondheim – Skogn – Meeting	248.5	3.5
4	Meeting – Orkanger - Meeting	293.3	42.7

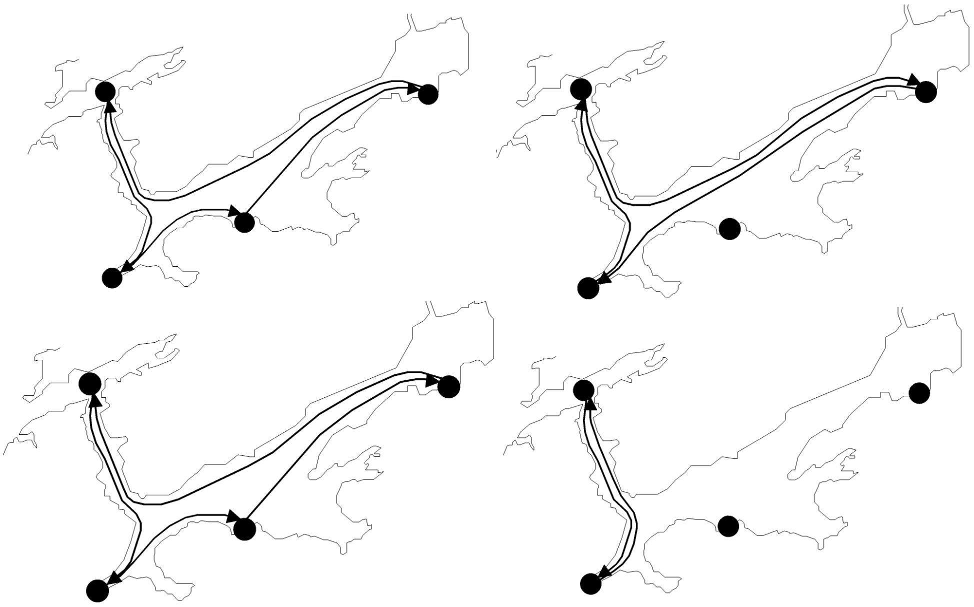


Figure 10: Illustration of routes generated from step 2 of optimization model. Expectation values of transportation demand and 24/7 port service.

The total sailing distance for one cycle for the routes constructed from step 2 of the optimization model is 411 NM, reduced by 16 NM compared to the routes found from step 1 of the optimization model.

7.4.2 Verification of simulation model

To verify that the simulation model run as intended and that the output is logged and processed in a correct manner, the simulation model is run with the same deterministic data as the optimization model, 100% operability of the ship-to-ship cargo transfer operation, and daughter vessel speed of 2.5 knots. The model is run for 200 cycles (400 weeks) where the time spent waiting for the mother vessel at the meeting point is logged for every period in every cycle. The mean, maximum and minimum waiting times are calculated for each period and presented in Table 14.

Table 14: Waiting times experienced in each period for deterministic model run with 100% operability of ship-to-ship cargo transfer operation and speed=2.5 knots.

Period #	Average waiting time [h]	Min. waiting time [h]	Max. Waiting time [h]
1	4.2	4.2	4.2
2	8.5	8.5	8.5
3	3.5	3.5	3.5
4	42.7	42.7	42.7

It is seen that the average, minimum and maximum waiting times experienced over the 100 cycles are the same as the waiting times found from the optimization model.

7.4.3 Schedule adherence calculated from simulation model runs

To evaluate the schedule adherence of the routes constructed from the optimization model, the simulation model is run for different levels of operability of the ship-to-ship cargo transfer operation and for a range of different speeds, starting from the speed found from the optimization model.

The model is first run for a deterministic scenario, where the number of containers to be loaded in each port per visit and the cargo transfer rates are the expectation values as used in the optimization model. Thereafter the model is run for a stochastic scenario, where the number of containers to be loaded in each port per visit follows the uniform distribution with variations of ± 10% from the average, and the cargo transfer rates follow the triangular distributions as described in Table 10.

Figure 11 shows the share of meetings with the mother vessel that the daughter vessel arrives on time, versus speed of the daughter vessel and operability of the ship-to-ship cargo transfer operation for the deterministic scenario. The same results from the stochastic scenario are presented in Figure 12.

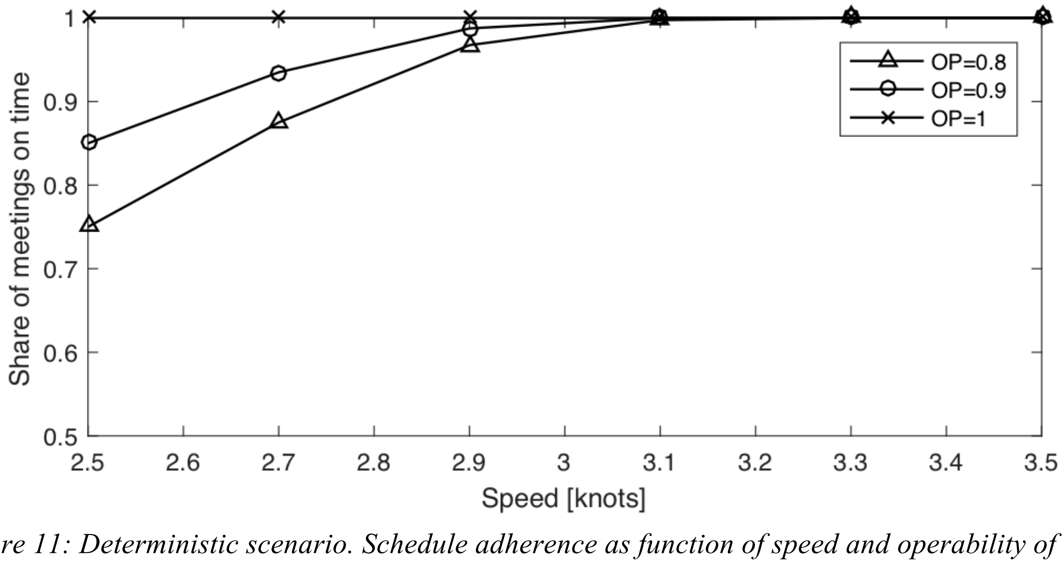


Figure 11: Deterministic scenario. Schedule adherence as function of speed and operability of ship-to-ship cargo transfer operation (OP). Routes designed for expectation values and 24/7 port service.

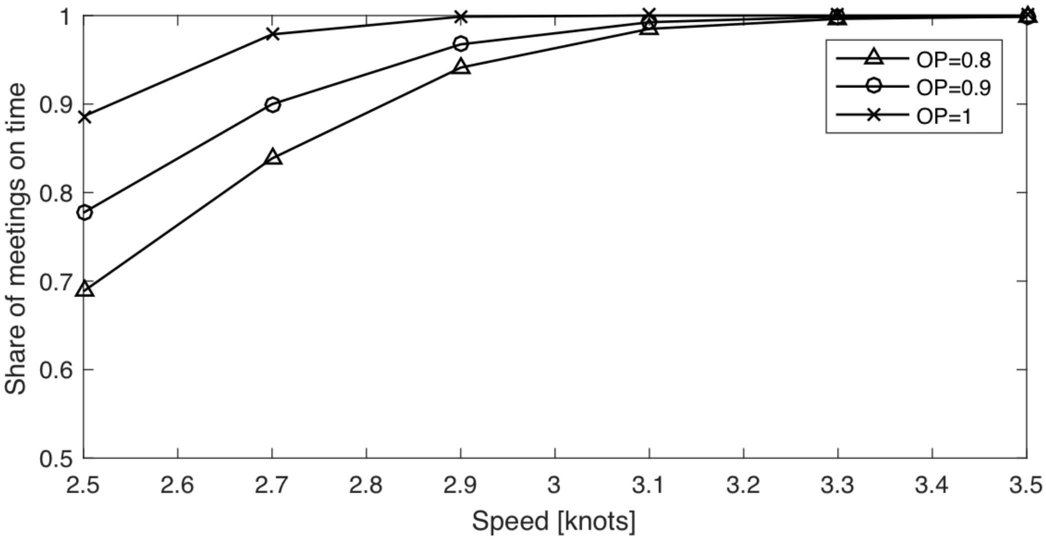


Figure 12: Stochastic scenario. Schedule adherence as function of speed and OP. Routes designed for expectation values and 24/7 port service.

For the deterministic scenario, when the operability of the ship-to-ship cargo transfer operation is 100% (OP=1), it is seen that the daughter vessel always arrives at the meeting point in time for speeds of 2.5 knots and above, which corresponds with the results from the optimization model. For OP=0.8 and 0.9 the speed of the daughter vessel must be increased to above 3 knots to achieve the same reliability.

For the stochastic scenario, when OP=1, it is seen that the speed of the daughter vessel must be increased to about 3 knots to always arrive at the meeting in time. For OP =0.9 and 0.8, the speed must be increased to about 3.3 knots to achieve the same reliability.

7.5 CaseA2 – Designing routes for 90th percentile of transportation demand and 24/7 port service

In this case the routes and minimum speed are again determined based on 24/7 opening hours in the ports. However, the number of containers to be loaded in each port is set to the 90th percentile for transportation demands, not the expectation values, when the routes are generated in the optimization model.

7.5.1 Routes generated from optimization model

The minimum speed found from step 1 of the model is 2.5 knots, equal to the minimum speed found in case 1. The routes found from step 2 of the optimization model are also identical to the routes found in case 1, with a sailing distance of 411 NM per cycle. This means that the routes determined in case 1 are still optimal when the transportation demand is increased to the

90th percentile in the optimization model. As the solution from this case is identical to case 1, no new simulations are carried out.

7.6 Case B1 – Designing routes for expectation values of transportation demand and port opening hours from 07.00 to 15.00

In this case, the routes and minimum speed are determined based on limited port opening hours from 07.00 to 15.00 and expectation values (mean) for transportation demands and cargo transfer rates.

7.6.1 Routes from optimization model

The minimum speed found as output from step 1 of the optimization model is 2.5 knots. The constructed routes found from step 2 of the optimization model are described in Table 15 and illustrated in Figure 13. The total sailing distance for one cycle is 413 NM.

Table 15: Description of routes from step 2 of optimization model. Expectation values of transportation demand and port opening hours from 07.00 to 15.00.

Period #	Route description
1	Meeting – Orkanger – Meeting
2	Meeting – Trondheim – Skogn – Orkanger – Meeting
3	Meeting – Orkanger – Skogn – Meeting
4	Meeting – Trondheim – Skogn – Orkanger - Meeting

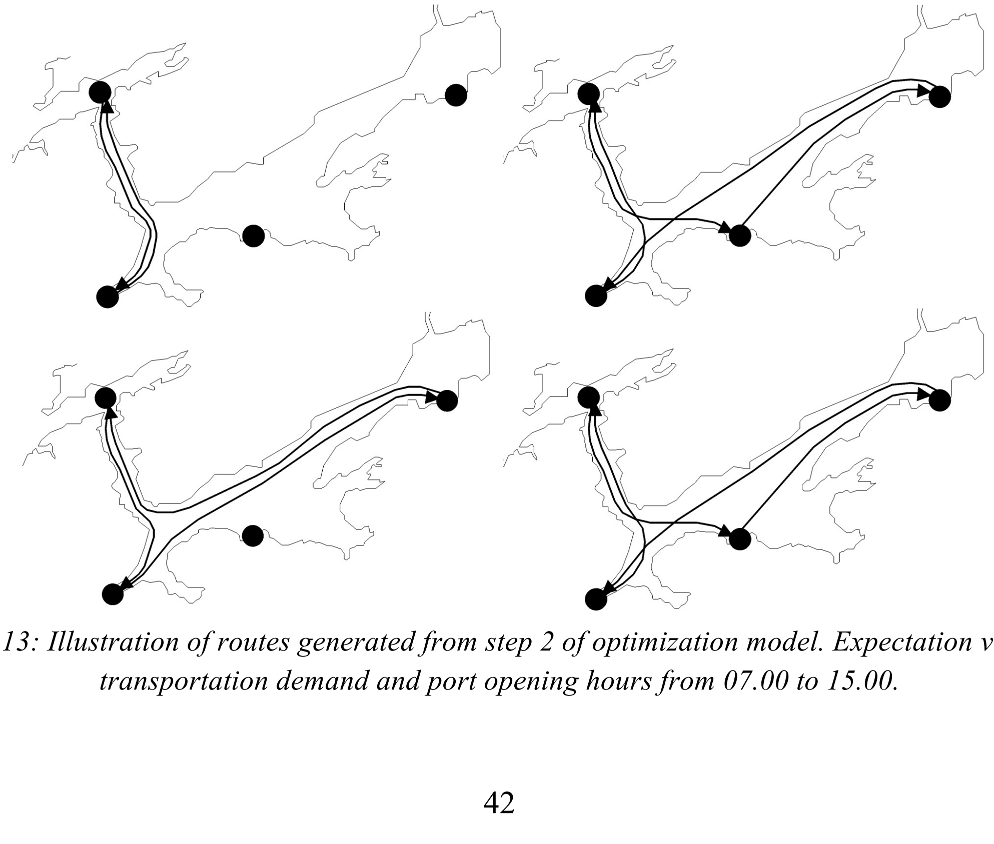


Figure 13: Illustration of routes generated from step 2 of optimization model. Expectation values of transportation demand and port opening hours from 07.00 to 15.00.

7.6.2 Schedule adherence calculated from simulation model runs

Figure 14 shows the share of meetings with the mother vessel that the daughter vessel arrives on time, versus speed of the daughter vessel and operability of the ship-to-ship cargo transfer operation for the deterministic scenario. The same results from the stochastic scenario are plotted in Figure 15.

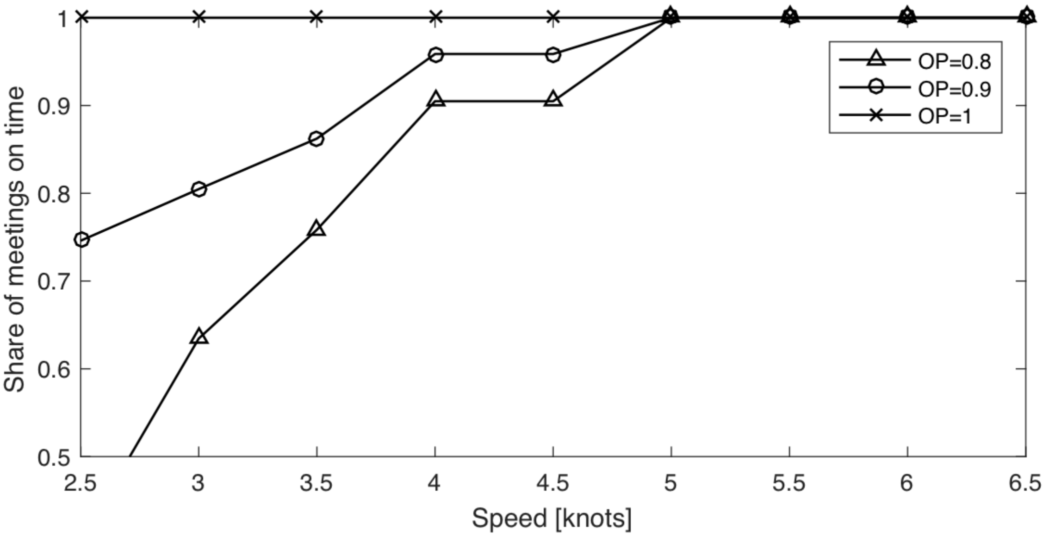


Figure 14: Deterministic scenario. Schedule adherence as function of speed and OP. Routes designed for expectation values of demand and 24/7 port service.

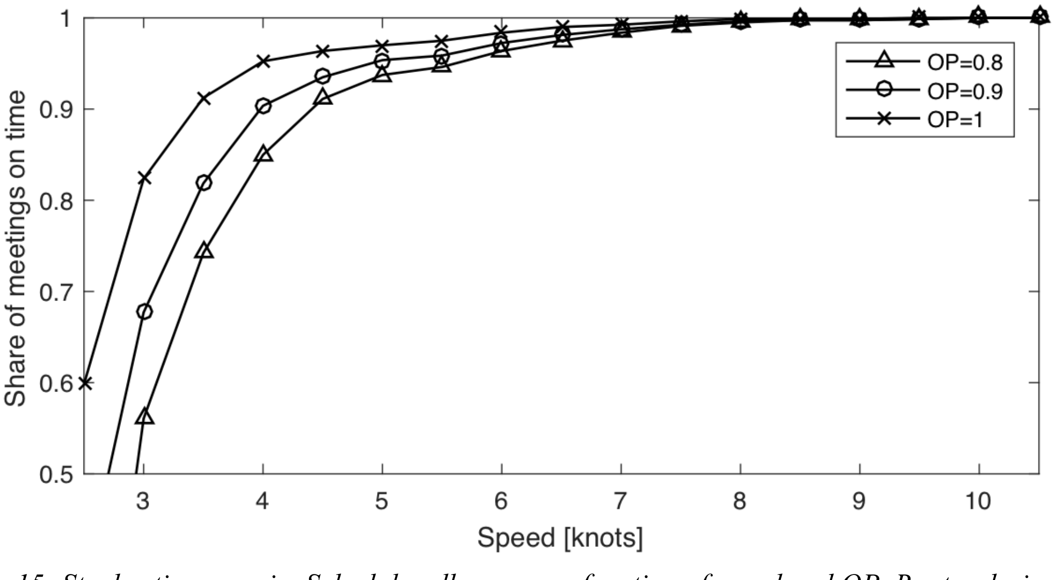


Figure 15: Stochastic scenario. Schedule adherence as function of speed and OP. Routes designed for expectation values of demand and 24/7 port service.

For the deterministic scenario, when OP=1, it is seen that the daughter vessel always arrives on time if the speed of the vessel is 2.5 knots or above. When OP=0.8 and 0.9 the vessel speed must be increased to about 5 knots to achieve the same reliability.

For the stochastic scenario, the speed must be increased to about 8 knots for all evaluated operabilities of the ship-to-ship cargo transfer operation to always arrive at the meeting point on time.

7.7 Case B2 – Designing routes for 90th percentile of transportation demand and port opening hours from 07.00 to 15.00

In this case the routes and minimum speed are again determined based on port opening hours from 07.00 to 15.00. However, the number of containers to be loaded/unloaded in each port are set to the 90th percentile of transportation demand, not the expectation values, when generating the routes in the optimization model.

7.7.1 Routes from optimization model

The minimum speed found from step 1 of the optimization model is 3 knots. The generated routes found when minimizing the sailing distance in the cycle, step 2 of the optimization model, are described in Table 16 and illustrated in Figure 16. The sailing distance for one cycle is 411 NM.

Table 16: Description of routes from step 2 of optimization model. 90th percentile of transportation demand and port opening hours from 07.00 to 15.00.

Period #	**Route description**
1	Meeting – Orkanger – Skogn - Meeting
2	Meeting – Orkanger – Trondheim – Skogn – Meeting
3	Meeting – Orkanger - Meeting
4	Meeting – Orkanger – Trondheim – Skogn - Meeting

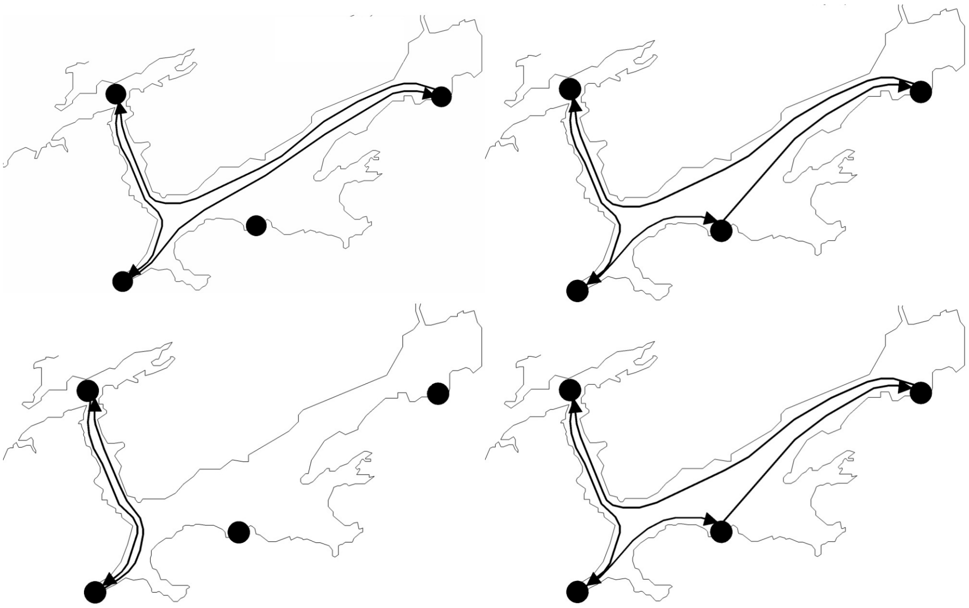


Figure 16: Illustration of routes generated from step 2 of optimization model. 90th percentile of transportation demand and port opening hours from 07.00 to 15.00.

7.7.2 Schedule adherence calculated from simulation model runs

Figure 17 shows the share of meetings with the mother vessel that the daughter vessel arrives on time, versus speed of the daughter vessel and operability of the ship-to-ship cargo transfer operation for the deterministic scenario. The same results from the stochastic scenario are plotted in Figure 18.

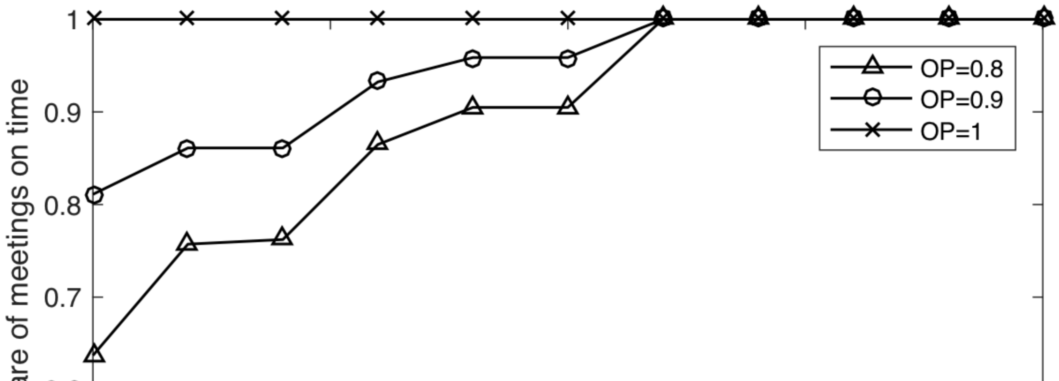


Figure 17: Deterministic scenario. Schedule adherence as function of speed and OP. Routes designed for 90th percentile of demand and port opening hours from 07.00 to 15.00.

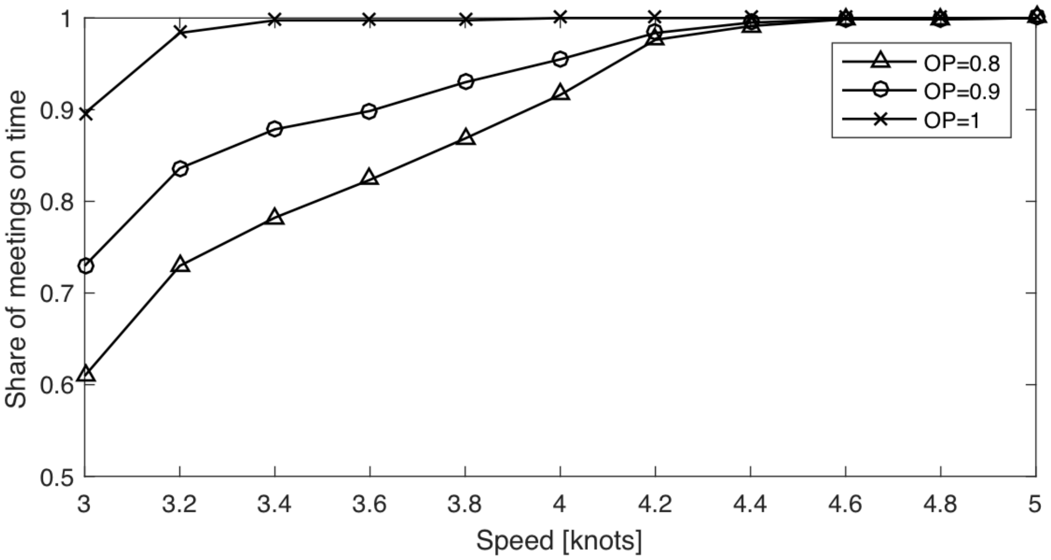


Figure 18: Stochastic scenario. Schedule adherence as function of speed and OP. Routes designed for 90th percentile of demand and port opening hours from 07.00 to 15.00.

For the deterministic scenario, when OP=1, it is seen that the daughter vessel always arrives on time for speeds of 3 knots or above. For OP=0.9 and 0.8 the speed must be increased to above 4 knots to achieve the same reliability.

For the stochastic scenario, when OP=1, the vessel always arrives on time for speeds of about 3.5 knots and above. For OP=0.9 and 0.8 the speed must be increased to about 4.5 knots to achieve the same reliability.

7.8 Comparative discussion of case studies

The simulation model is verified by running the model with deterministic input data and comparing results to the waiting times found from the optimization model. This is done for case A1, where waiting times in all periods are calculated and compared. For the other cases, when the simulation model is run for the deterministic scenario it is seen that the daughter vessel always arrives in time at the meeting point when the speed from the optimization model is used.

The main results from the case studies is presented in Table 17, where the simulation model results are from the stochastic scenario. The speeds in the table are the speeds required to always arrive at the meeting point in time for the mother vessel.

Table 17: Results from case studies.

Case	Distance travelled per cycle [NM]	Speed from optimization model [knots]	Speed from simulation model, OP=1 [knots]	Speed from simulation model, OP<1 [knots]
A1	411	2.5	3	3.3
A2	411	2.5	3	3.3
B1	413	2.5	8	8
B2	411	3	3.5	4.6

Minimum speed and routes found from the optimization model are identical for case A1 and A2 even though the time spent in port is higher in A2, where the 90th percentile of demands are used. The discretization of speeds with step-length of 0.5 knots results in slack for all periods in case A1. This is apparent from Table 13 where it is seen that the minimum time spent waiting for the mother vessel is 3.5 hours. This slack allows the vessel to travel with the same speed even though time in port is increased. In the stochastic scenario, for case A1 and A2, the speed of the daughter vessel must be increased to about 3.3 knots to always arrive at the meeting point in time. This corresponds to an increase of about 30% from the speed found as output from the optimization model.

For case B1, when restricted opening hours are introduced in the ports, results from the optimization model are not very different from case A1 and A2. The minimum speed from the optimization model is found to 2.5 knots, which is the same result as for A1 and A2, while the total sailing distance increase slightly by 2 NM. The system is hence little affected by opening hours when it is subject to a deterministic nature. However, when the routes are tested for the stochastic scenario, the speed of the daughter vessel must be increased to about 8 knots to always arrive in time for the mother vessel. This corresponds to an increase of about 220% from the speed found as output from the optimization model. For restricted opening hours and the routes found as output from case B1, the system is considerably more affected by variations in the system compared to 24/7 port service and the routes found in case A1 and A2.

For case B2, when the 90th percentile of transportation demands are used to construct routes for restricted opening hours, the minimum speed is found to 3 knots from the optimization model. The total sailing distance is slightly reduced compared to case B1, to the same value as for A1 and A2. For the stochastic scenario, with operability of 100%, the speed must be increased to about 3.5 knots to always arrive at the meeting point in time. For operabilities less than 100%, the speed must be increased to about 4.5 knots. By using the 90th percentile of transportation demand in the case of restricted opening hours, a more robust solution is hence established compared to the routes found in case B1.

Although the difference in results from the optimization model are small between the cases, the different solutions perform remarkably different when the system is subject to a stochastic nature. This accentuates the importance of considering uncertainties when designing the system, and to evaluate the robustness of considered solutions.

From all four cases, it is seen that the relative difference in reliability for different levels of operability of the ship-to-ship cargo transfer operation decrease with increasing speed. This is reasonable as relative time spent sailing decrease when speed increase. Relative time spent in port and waiting hence increase. This also explains the general diminishing nature of the plots. Increasing vessel speed will give lower marginal contribution to reliability for higher speeds.

7.9 Reduction in simulation time by storing Excel documents in persistent variables

To evaluate the effect of storing Excel sheets in persistent variables, as discussed in Chapter 6.2.4, the simulation model is run both with and without use of this procedure in the different system components for case B1. Both simulations are performed over 200 cycles (400 weeks) with speed of 2.5 knots and operability of ship-to-ship cargo transfer of 80%.

When Excel documents are stored in persistent variables the simulation time is 8 seconds, including compiling of the model. When the procedure for storing Excel sheets in persistent variables is removed from all system blocks, the simulation time increases to 12 minutes and 26 seconds.

Considering that all cases are run for 3 different levels of operability and minimum 6 different speeds, minimum 18 simulations is performed per case. Using persistent variables for accessing Excel documents hence decrease the expected simulation time from about 3 hours and 44 minutes to under 3 minutes per case for the smallest cases.

Case B1 is run for 16 different speeds and 3 different levels of operability, totaling in 48 simulations. For this case, the expected simulation time is decreased from about 10 hours to less than 7 minutes.

8. Discussion

Considering the objective of constructing decision support models that can be used for different areas, and the rough assumptions related to data for the computational study, this chapter only discusses the main trends and relationships that can be drawn from the results, not the numerical results themselves. A discussion of the underlying assumptions regarding the operating context of the system is also provided.

From the case studies, it is found that routes and speed determined in the optimization model are little affected by introducing restricted opening hours in ports compared to 24/7 port service. However, when evaluating the routes in the simulation model, where the system is subject to uncertainties, it is evident that the robustness of the system decrease when port opening hours are introduced. By designing the routes for the 90th percentile of transportation demands in the optimization model, more robust routes are generated for restricted port opening hours compared to the routes designed for the expectation values of demands.

Share of meetings that the daughter vessel arrives at the transshipment meeting point before the mother vessel are generally found to have diminishing return for increasing daughter vessel speeds. This is as expected as the relative time spent sailing decrease with increasing speed, while relative time in port and relative time spent waiting increase. One of the main ideas of the Short Sea Pioneer is to reduce this diminishing effect for the mother vessel by using the daughter vessel for time-consuming port operations.

As for operability of the ship-to-ship cargo transfer operation, the produced results clarify the importance of considering the duration of the operation. For operabilities less than 100%, it is evident that the expected operability decrease with increasing duration of the operation. This decreasing effect is more significant for the lower levels of operational criterion. It should be noted that the evaluated time series of significant wave heights are obtained from a forecast model provided by the Norwegian Meteorological Institute, and are not actual observed wave heights. In addition, the evaluated time series only contains data points for one year. For the higher levels of operational criterion, number of calms hence become small, which might influence the validity of the results. To increase the amount of data points, one possible method is to use Markov Chains. Bootstrapping or inverse transform sampling to extend the time series should be avoided as these methods do not consider the persistency of the weather.

In addition to the assumptions and limitations regarding input data for the computational study, several assumptions about the operating context for the vessels are also included in the models. These assumptions should be kept in mind if one wishes to utilize the models.

For restricted opening hours in ports, it is assumed that the daughter vessel only has to start loading/unloading within the opening hours and that it can finish this operation even if the duration of the operation exceeds the closing time of the port. The validity of this assumption should be questioned as port staff are probably needed to perform loading and unloading. It could be argued that this assumption is valid for automatic loading and unloading systems. However, a counterargument would then be the existence of opening hours if no personnel are needed for port operations.

Mother vessels are assumed to always arrive on time, and are hence modelled as fixed points in time in the models. Modelling mother vessel arrivals as fixed points in time excludes the effect of a delayed mother vessel arrival on the daughter vessel system. In the real world, if a mother vessel arrives late in a period, and the next mother vessel arrival is independent of the previous mother vessel arrival, the daughter vessel will have less available time to complete the feeder route for that period.

It is assumed that the daughter vessel always can accommodate the cargo arriving on the mother vessel and in the ports, and hence load management are not included in the models. This assumption should be accounted for when using the models by either adjusting the cargo in ports to quantities that can be accommodated by the daughter vessel or by introducing load management and capacity constraints in the models. Capacity constraints are not included in the constructed models as the presence of capacity constraints also requires information of priority rules between the different ports. It could be argued that the lack of capacity constraints will never reduce the robustness of the solutions, when only considering arrival times at the meeting point, as time spent in port and at the transshipment meeting point will either remain unchanged or decrease when capacity constraints are implemented. However, the absence of capacity constraints might result in over-estimation of required sailing speeds and an unrealistic feeder capacity.

Required number of visits for the different ports are input to the optimization model, not determined by the solution. This can be convenient as some ports might have time-sensitive cargo and should therefore be visited frequently even though the volumes are not very high.

Related to previous work, the constructed optimization model has some commonalities with the fleet composition and supply ship routing model constructed by Halvorsen-Weare et al. (2012). The meeting point for transshipment operations could be thought of as the depot from the supply ship model, while the ports could be thought of as offshore installations. In both models the required number of visits for the ports/offshore installations are input to the model, and both models seek to determine cyclical routes that repeat themselves over the planning horizon. Opening hours in ports/offshore installations are also considered in both models.

Whereas the model proposed by Halvorsen-Weare et al. (2012) is constructed as a path-flow model with a priori generation of voyages, the model constructed in this thesis is an arc-flow model where the routes are constructed in the optimization model. Similar as Halvorsen-Weare et al. (2012) force slack in the solution by setting an upper limit of voyage duration, this can be done in the constructed optimization model by decreasing the end time of each period. In the case study, slack is assured for the average scenario by increasing the number of containers to be handled in port from the average level.

Although the optimization model does not present a more novel way of assuring robustness in the solution compared to Halvorsen-Weare et al. (2012), robustness of the different routes are evaluated in the simulation model, where the speed is adjusted to assure a more robust solution.

Compared to Agra et al. (2013) that specify the start and the end of the time window for each period, the constructed optimization model presents a more generic way of modelling port opening hours for arc-flow models. Modelling time windows as done in this thesis however requires that a port opens and closes at the same time every day.

9. Conclusion

This thesis set out to construct decision support models for designing daughter vessel feeder routes for the mother-daughter vessel system Short Sea Pioneer. An optimization model, a discrete event simulation model and a script to estimate operability of the ship-to-ship cargo transfer operation has been made and applied on a constructed feeder problem for the Trondheim Fjord.

The optimization model is configured to determine daughter vessel minimum sailing speed and to design cyclical routes over a given planning horizon based on a required number of visits for the involved ports. The simulation model is configured to evaluate the robustness of the constructed routes when the system is subject to uncertainties.

Input data for the evaluated cases are based on rough assumptions which downgrades the validity of the results. Although the numerical results are of low significance, the computational study shows that the models work as intended and are applicable for the problem nature. By comparing results from the different cases, it is also possible to gain insight into the system performance and how it is affected by changes in the operating context.

Combined use of the optimization and simulation model highlights the importance of considering uncertainties when designing the system. It should be noted that the models are based on several assumptions related to the operating environment and do in no way provide a perfect representation of the reality.

10. Further work

To increase the credibility of the models, stakeholders in the project and system experts should review and revise the assumptions and simplifications made about the operating context. Stakeholders should also be involved in the formulation of objective functions to make sure that the constructed routes better meet their requirements.

For further expansion of the constructed models it is recommended to model port operations and ship-to-ship cargo transfer operation in greater detail as considerable time is spent in port and at the meeting point with the mother vessel. It is also recommended to model the flow of containers in greater detail and to extend the models to include load management and capacity constraints.

To capture the effect of late arrivals of mother vessels on the daughter vessel feeder system, it is recommended to give more attention to the arrivals of mother vessels in the model. This could either be done by including operation of the mother vessels in the model, or by modelling the arrival times of the mother vessels as a distribution. To include variations in mother vessel arrival times, information about the routes and operating environment of the mother vessels are needed.

Different routes for the different periods results in high idle time/time spent waiting for the mother vessel in some of the periods. It would be of interest to investigate the possibility of using the daughter vessel for alternative services in these periods, for instance relocation of empty containers from import ports to export ports in the area.

References

Agra, A., Christiansen, M., & Delgado, A. (2013). Mixed Integer Formulations for a Short Sea Fuel Oil Distribution Problem. *Transportation Science, 47*(1), 108-124. doi:10.1287/trsc.1120.0416

Banks, J., Nelson, B. L., & Carson, J. S. (2005). *Discrete-event system simulation* (4th ed. ed.). Upper Saddle River, N.J: Pearson/Prentice-Hall.

Caris, A., Macharis, C., & Janssens, G. K. (2011). Network analysis of container barge transport in the port of Antwerp by means of simulation. *Journal of Transport Geography, 19*(1), 125-133. doi:10.1016/j.jtrangeo.2009.12.002

Carson. (2003). Introduction to modeling and simulation (Vol. 1, pp. 7-13). USA.

Chong, E. K. P., & çZak, S. H. (2014). *An Introduction to Optimization* (Fourth edition. ed.): United States: John Wiley & Sons Inc.

Cigolini, R., Pero, M., Rossi, T., & Sianesi, A. (2014). Using simulation to manage project supply chain in the offshore oil and gas industry. *Production Planning & Control*, 1-11. doi:10.1080/09537287.2013.870360

European Commission. Directorate-General for Mobility, a. (2011). *White paper on transport : roadmap to a single European transport area : towards a competitive and resource-efficient transport system*

Fazi, S., Fransoo, J. C., & Van Woensel, T. (2015). A decision support system tool for the transportation by barge of import containers: A case study. *Decision Support Systems, 79*, 33-45. doi:10.1016/j.dss.2015.08.001

Halvorsen-Weare, E. E., Fagerholt, K., Nonås, L. M., & Asbjørnslett, B. E. (2012). Optimal fleet composition and periodic routing of offshore supply vessels. *European Journal of Operational Research, 223*(2), 508-517. doi:10.1016/j.ejor.2012.06.017

Hillier, F. S., & Lieberman, G. J. (2015). *Introduction to operations research* (10th ed. ed.). New York: McGraw-Hill.

Hu, T. C., & Kahng, A. B. (2016). *Linear and Integer Programming Made Easy*. Cham: Springer International Publishing: Cham.

Huang, Y.-F., Hu, J.-K., & Yang, B. (2015). Liner services network design and fleet deployment with empty container repositioning. *Computers & Industrial Engineering, 89*, 116-124. doi:10.1016/j.cie.2015.01.021

Institute of Transport Economics. (2015). *Potential and measures for transfer of freight from road to sea transport* Vol. 1424/2015. (pp. 64).

Kim, J., & Morrison, J. R. (2012). Offshore port service concepts: classification and economic feasibility.(Report). *Flexible Services and Manufacturing Journal, 24*(3), 214.

Kongsberg. (2017). Autonomous Shipping. Retrieved from https://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/597733F8A1B8C640C12580AC0049C134?OpenDocument 27.05 2017

Konings, R., Van Der Horst, M., Hutson, N., & Kruse, J. (2010). Comparative strategies for developing hinterland transport by container barge analysis for Rotterdam and U.S. ports. *Transportation Research Record*(2166), 82-89. doi:10.3141/2166-10

Larsen, K. (2017). *Marine Operations, Lecture notes NTNU*

Leach, P. T. (2007). End of the LASH era. *The Journal of Commerce*, 16.

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

MathWorks. (2017a). Retrieved from https://se.mathworks.com/products/simevents.html 05.02 2017

MathWorks. (2017b). Documentation. Retrieved from https://se.mathworks.com/help/matlab/ref/global.html 31.05 2017

MathWorks. (2017c). Persistent Variables. Retrieved from https://se.mathworks.com/help/hdlcoder/ug/persistent-variables-1.html 31.05 2017

Monios, J. (2017). Cascading feeder vessels and the rationalisation of small container ports. *Journal of Transport Geography, 59*, 88-99. doi:10.1016/j.jtrangeo.2017.02.001

NCA. (2017). Retrieved from http://kart.kystverket.no/ 02.02 2017

Nielsen, F. G. (2008). *Marine Operations, Lecture notes NTNU.*

Portofrotterdam. Inland shipping. Retrieved from https://www.portofrotterdam.com/en/connections-logistics/intermodal-transportation/inland-shipping 03.03 2017

Samferdselsdepartement, D. K. (2013). *Meld.St.26 Nasjonal transportplan 2014 – 2023.*

SSB. (2016). Maritime transport. Retrieved from http://ssb.no/en/transport-og-reiseliv/statistikker/havn/kvartal/2016-12-12 - content 04.03 2017

Tang, O., Wong, P., & Lau, Y. Y. (2013). An Economic Study of Mid-Stream Operations in Hong Kong. *Seaview 2013.*

TrondheimPortAuthority. (2016). Retrieved from http://trondheimhavn.no/levanger.aspx 03.04 2017

Appendices

A. Task description

MASTER'S THESIS IN MARINE TECHNOLOGY

SPRING 2017

For stud.techn.

Hans Henrik Nordheim

Designing Feeder Routes in a Mother-Daughter Vessel System with Ship-to-Ship Cargo Transfer

A Combined Optimization – Simulation Approach

Background

There is a high political pressure to transfer goods from road to sea transportation. The EU have set a clear and ambitious goal of shifting 30% of the goods that today are transported by trucks on distances longer than 300 km to rail or sea transportation by 2020. The corresponding goal for 2050 is 50%. The Norwegian government have also announced that they will support measures to increase the competitiveness of short sea shipping solutions.

Short Sea Pioneer is a next generation short sea container shipping system that is currently under development by NCE Maritime CleanTech. The system is developed to solve some of the challenges currently faced by the short sea shipping industry and consists of two different types of vessels: mother vessels as distributors and daughter vessels as feeders. The mother vessels will be used to transport containers on relatively long distances and only visit larger ports along it's roundtrips. The daughter vessels will be used in areas where there is a cluster of ports, and will be used to feed containers from the mother vessels to the ports in the area, and vice versa. Cargo transfer between mother and daughter vessels will find place at sea.

By utilizing daughter vessels to feed containers one can access a larger number of ports and at the same time decrease the number of port calls for the large mother vessels. Hence it is believed that the mother daughter vessel system will decrease the total roundtrip time which in turn can allow for more frequent departures. Using daughter vessels to access smaller ports can also lead to reduction in truck activity and port congestion in the larger ports.

Although the mother-daughter vessel system Short Sea Pioneer presents potential benefits, it also poses a higher level of complexity by adding more elements to the system. As Short Sea Pioneer will operate as a liner shipping system, the vessels will need to follow fixed pre-published schedules. The mother vessel system and the daughter vessel system will therefore need to be precisely coordinated.

Objective

The objective of this thesis is to develop decision support models for designing daughter vessel feeder routes in the mother-daughter vessel system Short Sea Pioneer to ensure that each port is visited its desired number of times and that the daughter vessel do not delay the mother vessels.

The models will be applied on a constructed feeder case for the Trondheim area to demonstrate potential application areas of the models.

Tasks

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

a. Review state of art within the topic. That means to document what others have done and published previously.
b. Construct an optimization model that can be used to design daughter vessel routes that can be travelled with the lowest possible speed.
c. Construct a simulation model that can be used to assess the performance of the routes generated from the optimization model when the system is subject to uncertainties.
d. Construct a script to estimate the operability of the ship-to-ship cargo transfer operation based on operational criterion and duration of the operation.
e. Apply the models on a constructed case for the Trondheim Fjord
f. Provide recommendations for further work.

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.

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:

Company contact: Kay Fjørtoft, SINTEF Ocean

Deadline: 11.06.2017

Bjørn Egil Asbjørnslett
Professor/Main Supervisor

B. Optimization model as implemented in Xpress

Xpress model:

```
model FeederNetwork

options explterm

options noimplicit

uses "mmxprs";

! Specify data file
parameters
      DataFile = 'Input.txt';
end-parameters

!declare parameters:
declarations
      nPorts:         integer;
      nTimePeriods: integer;
      nVessels: integer;
      Ports:          set of integer;
      TimePeriods: set of integer;
      Vessels: set of integer;
      Distances:           array(Ports, Ports)  of real;
      TotalVisits:    array(Ports) of integer;
      Cargoes:   array(Ports) of integer;
      LoadingRates: array(Ports) of integer;
      StartTimePeriods: array(TimePeriods) of real;
      EndTimePeriods: array(TimePeriods) of real;
      Speeds:    array(Vessels) of real;
      TimeMooringMeeting: real;
      TimeMooringPort: real;
      OpeningHourPorts: real;
      ClosingHourPorts: real;

end-declarations

!Initialize parameters from data file
initializations from DataFile
      nPorts;
      nTimePeriods;
      nVessels;
      TimeMooringMeeting;
      TimeMooringPort;
      OpeningHourPorts;
      ClosingHourPorts;
end-initializations

!Define the sets based on the number of "facilities":
Ports := 1 .. nPorts;
TimePeriods := 1 .. nTimePeriods;
Vessels := 1 .. nVessels;

finalize(Ports);
finalize(TimePeriods);
finalize(Vessels);

!Initialize parameters from data file
initializations from DataFile
      Distances;
      TotalVisits;
      StartTimePeriods;
      EndTimePeriods;
      Speeds;
      Cargoes;
      LoadingRates;

end-initializations

!Declare variables:
! Dynamic arrays are used to declare indexed variables.
! All variables are by default >= 0
declarations
      Use:            dynamic array(Vessels) of mpvar;
      Flow:          dynamic array(Vessels, Ports, Ports, TimePeriods)
            of mpvar;
      Time:           dynamic array(Ports, TimePeriods) of mpvar;
      LoadMeeting: dynamic array(TimePeriods) of mpvar;
      UnloadMeeting: dynamic array(TimePeriods) of mpvar;
      VisitDay: dynamic array(Ports, TimePeriods) of mpvar;

end-declarations

forall (vv in Vessels, ii in Ports, jj in Ports | jj<>ii, pp in
TimePeriods) do
```

```
     create(Flow(vv,ii,jj,pp));
     Flow(vv,ii,jj,pp) is_binary;
end-do

forall(vv in Vessels) do
     create(Use(vv));
     Use(vv) is_binary;
end-do

forall (ii in Ports, pp in TimePeriods) do
     create(Time(ii,pp));
end-do

forall (pp in TimePeriods) do
     create(LoadMeeting(pp));
     create(UnloadMeeting(pp));
end-do

forall (ii in Ports, pp in TimePeriods) do
     create(VisitDay(ii,pp));
     VisitDay(ii,pp) is_integer;
end-do

declarations
     Speed:    linctr;
     TotalDistance: linctr;
end-declarations

!OBJECTIVE FUNCTIONS:
!OBJECTIVE FUNCTION STEP 1
Speed :=
     sum(vv in Vessels)Speeds(vv)*Use(vv);

!OBJECTION FUNCTION STEP 2
TotalDistance:=
     sum(vv in Vessels, ii in Ports, jj in Ports, pp in
TimePeriods)Distances(ii,jj)*Flow(vv,ii,jj,pp);

!CONSTRAINTS:
!Make sure that decision variable is 1 if vessel v is used:
forall(vv in Vessels, pp in TimePeriods) do
     Use(vv)-sum(jj in Ports)Flow(vv,1,jj,pp)>=0;
end-do
```

```
!Make sure that only 1 vessel is used:
!NB! ONLY NECESSARY FOR STEP 2
sum(vv in Vessels)Use(vv)=1;

!Assure sufficient number of visits for eacg port during planning
horizon:
forall(ii in Ports | ii<>1 and ii<>5) do
     sum(vv in Vessels, jj in Ports, pp in
TimePeriods)Flow(vv,ii,jj,pp)=TotalVisits(ii);
end-do

!Flow conservation:
forall(vv in Vessels, jj in Ports| jj<>1 and jj<>5, pp in TimePeriods)
do
     sum(ii in Ports)Flow(vv,ii,jj,pp)-sum(ii in
Ports)Flow(vv,jj,ii,pp)=0;
end-do

! START FROM ORIGIN. END AT DESTINATION. CAN NOT START FROM DESTINATION
OR END AT ORIGIN
forall(pp in TimePeriods) do
     sum(vv in Vessels, jj in Ports)Flow(vv,1,jj,pp)=1;
     sum(vv in Vessels, ii in Ports)Flow(vv,ii,1,pp)=0;

     sum(vv in Vessels, ii in Ports)Flow(vv,ii,5,pp)=1;
     sum(vv in Vessels, jj in Ports)Flow(vv,5,jj,pp)=0;
end-do

!Amount unloaded and loaded at meeting:
forall(pp in TimePeriods) do
     LoadMeeting(pp)=sum(vv in Vessels,ii in Ports | ii<>1, jj in
Ports)Cargoes(ii)*Flow(vv,ii,jj,pp);
end-do

forall(pp in TimePeriods | pp<>1) do
     UnloadMeeting(pp)=LoadMeeting(pp-1);
end-do

UnloadMeeting(1)=LoadMeeting(4);

!Set start time for each period:
forall (pp in TimePeriods) do
Time(1,pp)=StartTimePeriods(pp);
end-do
```

```
!UPDATE TIME:
forall(vv in Vessels, ii in Ports | ii<>1, jj in Ports | jj<>1, pp in
TimePeriods) do
     Time(jj,pp)-Time(ii,pp)-Distances(ii,jj)/Speeds(vv)-
TimeMooringPort-
(Cargoes(ii)*2)/LoadingRates(ii)+EndTimePeriods(pp)*(1-
Flow(vv,ii,jj,pp))>=0; !EndTimePeriods(pp)=M
!    Force time to exact value:
!    REMOVED FOR MODEL RUN WITH PORT OPENING HOURS
!    Time(jj,pp)-Time(ii,pp)-Distances(ii,jj)/Speeds(vv)-
TimeMooringPort-(Cargoes(ii)*2)/LoadingRates(ii)-
EndTimePeriods(pp)*(1-Flow(vv,ii,jj,pp))<=0;

end-do

forall(vv in Vessels, jj in Ports | jj<>1, pp in TimePeriods) do
     Time(jj,pp)-Time(1,pp)-Distances(1,jj)/Speeds(vv)-
TimeMooringMeeting-
(UnloadMeeting(pp)+LoadMeeting(pp))/LoadingRates(1)+EndTimePeriods(p
p)*(1-Flow(vv,1,jj,pp))>=0;
     !    Force time to exact value:
!    REMOVED FOR MODEL RUN WITH PORT OPENING HOURS
!    Time(jj,pp)-Time(1,pp)-Distances(1,jj)/Speeds(vv)-
TimeMooringMeeting-
(UnloadMeeting(pp)+LoadMeeting(pp))/LoadingRates(1)-
EndTimePeriods(pp)*(1-Flow(vv,1,jj,pp))<=0;
end-do

!A port can only be visited in a period after the period is started
and before it is finished:
forall (ii in Ports, pp in TimePeriods) do
Time(ii,pp) >= StartTimePeriods(pp);
Time(ii,pp) <= EndTimePeriods(pp);
end-do

!MAKE SURE THAT PORT 3 IS NOT VISITED TWO TIMES IN A ROW:
forall (pp in TimePeriods | pp<>4) do
     sum(vv          in          Vessels,          jj          in
Ports)(Flow(vv,3,jj,pp)+Flow(vv,3,jj,pp+1))<=1;
end-do

sum(vv in Vessels, jj in Ports)(Flow(vv,3,jj,1)+Flow(vv,3,jj,4))<=1;
```

```
!Opening hours in ports:
forall(ii in Ports | ii<>1 and ii <>5, pp in TimePeriods) do
     Time(ii,pp)-VisitDay(ii,pp)*24<=ClosingHourPorts;
     Time(ii,pp)-VisitDay(ii,pp)*24>=OpeningHourPorts;

!Bind the variable VisitDay to Time
     VisitDay(ii,pp)<=Time(ii,pp)/24;
     VisitDay(ii,pp)>=Time(ii,pp)/24-1;
end-do

!WHICH OBJECTIVE FUNCTION TO MINIMIZE:
minimize(Speed);!STEP 1
!LOCK VESSEL FOR STEP 2:
!Use(2)=1;
!minimize(TotalDistance); !STEP 2

!Write optimal objective value to results:
writeln('Optimal objective value : ', getobjval);
! getobjval is used to retreive the optimal objective value

! Write what speed is used
writeln;
forall (vv in Vessels | getsol(Use(vv)) >= 0.01) do
     writeln('Vessel speed=', strfmt(getsol(Speeds(vv)),4),' knots');
end-do

!Write which vessel # is used
writeln;
forall (vv in Vessels | getsol(Use(vv)) >= 0.01) do
     writeln('Vessel number=', vv);
end-do

!Write routes for each period
writeln;
forall (pp in TimePeriods, ii in Ports, jj in Ports, vv in Vessels |
getsol(Flow(vv,ii,jj,pp)) > 0.01) do
     writeln(' Time period ',pp, ': ',
strfmt(getsol(Flow(vv,ii,jj,pp)),4), ' sails from ',ii, ' to  ', jj,
'. Arrives at t=' ,getsol(Time(jj,pp)),
'Day=' ,getsol(VisitDay(jj,pp)));
end-do

end-model
```

Input file for Xpress model:

```
!Input data

nTimePeriods : 4
nPorts : 5
nVessels: 2
TimeMooringMeeting: 2
TimeMooringPort:1
OpeningHourPorts:7
ClosingHourPorts:15

Distances : [
0             24.21  25.31  49.66  0             !Transshipment meeting point
24.21  0             19.02  44.44  24.21  !Orkanger
25.31  19.02  0             29.36  25.31  !Trondheim
49.66  44.44  29.36  0             49.66  !Skogn
0             24.21  25.31  49.66  0             !Transshipment meeting point
]

TotalVisits: [0 4 2 3 0]
Cargoes: [0 92 12 23 0]
LoadingRates: [33 15 10  10 33]

StartTimePeriods: [6 90 174 258]
EndTimePeriods:      [90 174 258 342]

Speeds:       [2 2.5]
```

C. SimEvents model including global variables and Simulink functions:

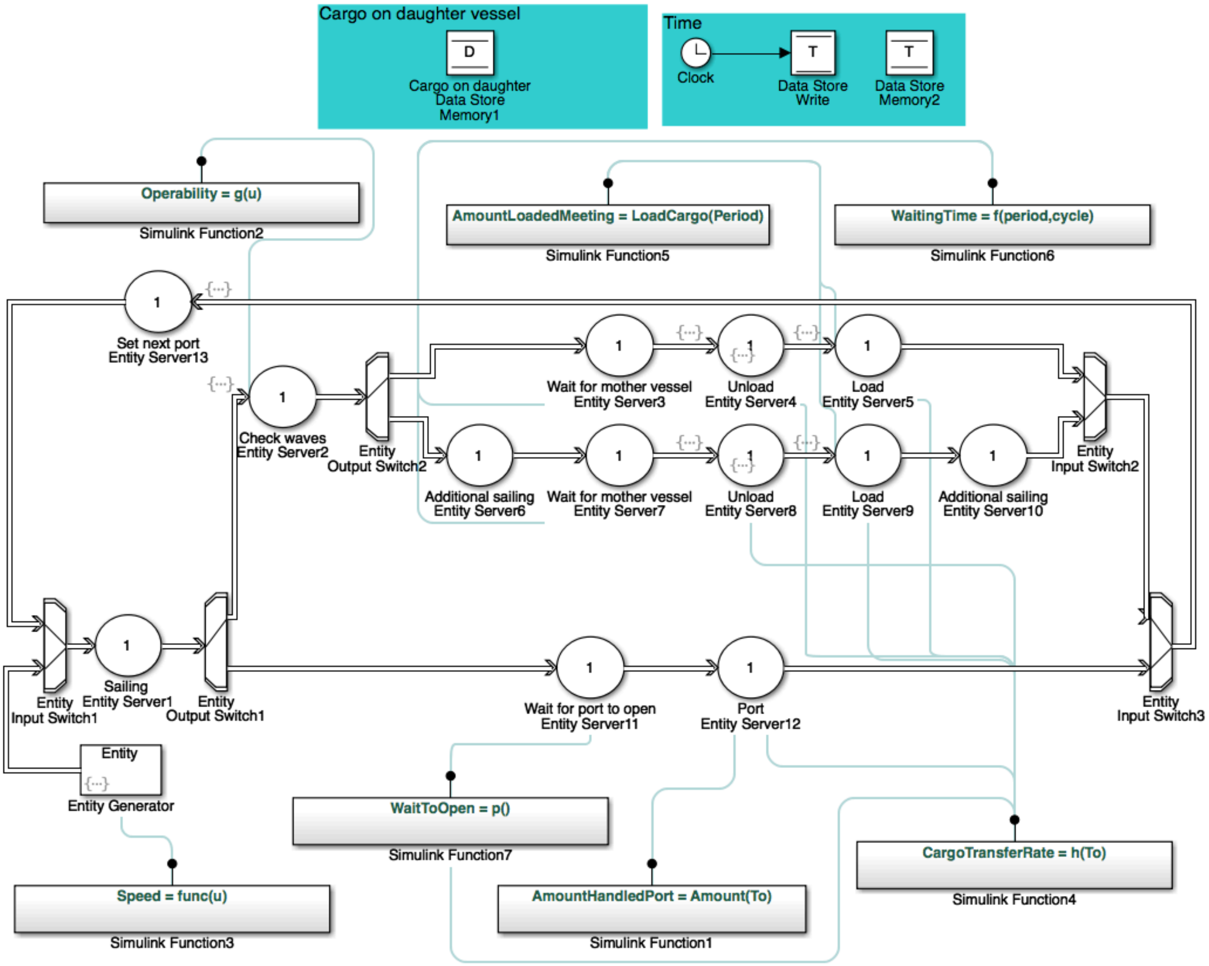


Figure 19: C. Simulation model from SimEvents including Simulink functions and global variables

MATLAB coding in the different system components:

Entity generator

<u>Tab: Entity generation</u>

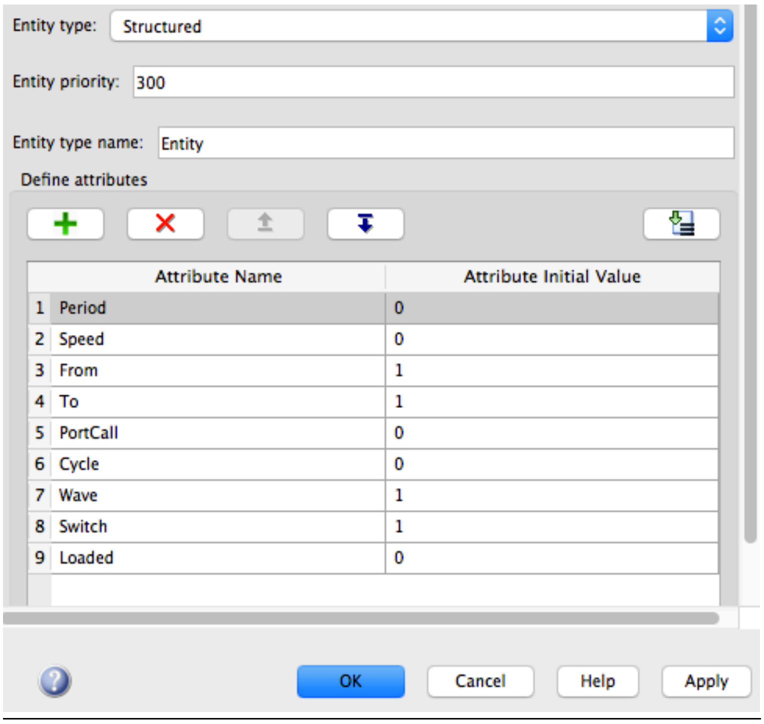
Generation method: Time-based

Time source: MATLAB action

Intergeneration time action:

```
%declare persistent variables:
persistent count igt
%xlsread not supported for stand alone code generation
%in SimEvents. Must be declared as extrinsic function:
coder.extrinsic('xlsread')

%Store intergeneration times in persistent vector:
if isempty(count)
    fleet=zeros(36,13);
    fleet=xlsread('Fleet');
    igt=fleet(1:2,1);
    count=1;
end
dt=igt(count);
count=count+1;

%NB! Make sure that Generate entity at simulation start
%box is not checked
```

Tab: Entity type



Entity type: Structured

Entity priority: 300

Entity type name: Entity

Define attributes

	Attribute Name	Attribute Initial Value
1	Period	0
2	Speed	0
3	From	1
4	To	1
5	PortCall	0
6	Cycle	0
7	Wave	1
8	Switch	1
9	Loaded	0

Figure 20: C. Configuration entity generator. Tab: Entity structure

Tab: Event actions

Event actions: Generate

```
%Call on Simulink function 3 to update the attribute Speed (defined
in script for running the model)
entity.Speed=func(1);
```

Entity server 1:

Tab: Main

Capacity: 1

Service time source: MATLAB action

Service time action:

```
%Declare distance matrix as persistent matrix so that it can be used
again without using the xlsread function
persistent DistMatrix
coder.extrinsic('xlsread')

if isempty(DistMatrix)
    %Initialize size of matrix:
    DistMatrix=zeros(6,4);
    %Read distance matrix from Excel:
```

```
    DistMatrix=xlsread('Distances');
end
%Find distance to be sailed by accessing element in distance matrix:
Distance=DistMatrix(entity.From,entity.To);

%Calculate sailing time (time spent in server)
dt=Distance/entity.Speed;
```

Entity output switch 1:

Number of output ports: 2
Switching criterion: From attribute
Switch attribute name:
Switch

Entity server 2:

Tab: Main
Capacity: 1
Service time source: Dialog
Service time value: 0
Tab: Event actions
Event actions: Entry

```
%Call on Simulink function 2 to check level of operability for ship-
to-ship cargo transfer operation (defined in script for running the
model)
Operability=g(1);

%random number between 0 and 1:
r=rand();

%Check if operation can find place at sea based on r and level of
operability:
if r<=Operability
    entity.Wave=1; %Can find place at sea
else
    entity.Wave=2; %Must sail to sheltered port
end
```

Entity output switch 2:

Number of output ports: 2
Switching criterion: From attribute
Switch attribute name:
Wave

Entity server 3:

Tab: Main
Capacity: 1
Service time source: MATLAB action
Service time action:

```
%Call on Simulink function 6 to calculate time spent waiting for
mother vessel:
%No waiting if period=0 (daughter vessel start operation from the
meeting point when deployed in the model)
if entity.Period >=1
```

```
    WaitingTime=f(entity.Period,entity.Cycle);
else
    WaitingTime=0;
end

dt = WaitingTime;
```

Tab: Event actions

Entity server 4:

Tab: Main

Capacity: 1

Service time source: MATLAB action

Service time action:

```
% Call Simulink function 4 to obtain cargo transfer rate:
CargoTransferRate=h(entity.To);

%Calculate time used for unloading to mother vessel (pluss time for
mooring/unmooring)
dt=entity.Loaded/CargoTransferRate+2;
```

Tab: Event actions

Event actions:

Entry

Entry action:

```
%UPDATE SCHEDULE
%Meeting mother vessel means start of next period
persistent c fleet nPeriods
coder.extrinsic('xlsread')

if isempty(c)
    fleet=zeros(36,13);
    fleet=xlsread('Fleet');
    nPeriods=fleet(2,7);
    c=1;
end

if entity.Period<nPeriods
    entity.Period=entity.Period+1; %Start on next period
else %If vessel completed last period
    entity.Period=1;
    entity.Cycle=entity.Cycle+1; %Start on first period
end

entity.PortCall=0; %Reset PortCall as vessel is starting to sail new
route
```

Tab: Event actions

Event actions:

Service complete

Service complete actions:

```
%Unloading completed. No cargo left on daughter vessel
entity.Loaded=0;
```

Entity server 5:

Tab: Main

Capacity: 1

Service time source: MATLAB action

Service time action:

```
%Cargo transfer rate:
CargoTransferRate=h(entity.To);

%Calculate time used for loading cargo from mother vessel (attribute
Loaded is updated in Event actions):
dt=entity.Loaded/CargoTransferRate;
```

Tab: Event actions

Event actions:

Entry

Entry action:

```
%Call Simulink function 5 to obtain number of TEUs to be loaded
%Update attribute Loaded:
entity.Loaded=LoadCargo(entity.Period);
```

Entity server 6:

Tab: Main

Capacity: 1

Service time source: MATLAB action

Service time action:

```
%Obtain additional sailing distance to sheltered meeting point from
Excel document.
%Make sure that xlsread function is maximum used once by use of
persistent variables and if statement:
persistent AddDist
coder.extrinsic('xlsread')
if isempty(AddDist)
    DistMatrix=zeros(6,4);
    DistMatrix=xlsread('Distances');
    AddDist=DistMatrix(6,4)
end
%Calculate additional time used for sailing to sheltered port
dt = AddDist/entity.Speed;
```

Entity server 7:

Identical to entity server 3. Waiting for mother vessel.

Entity server 8:

Identical to entity server 4. Unloading to mother vessel.

Entity server 9:

Identical to entity server 5. Loading from mother vessel.

Entity server 10:

Identical to entity server 6. Additional sailing from sheltered meeting port.

Entity server 11:

Tab: Main

Capacity: 1

Service time source: MATLAB action

Service time action:

```
%Call on Simulink function 7 to calculate time until port opens:
WaitToOpen=p();

%Wait outside of port until it opens before cargo transfer
operations are started:
dt = WaitToOpen;
```

Entity server 12:

Tab: Main

Capacity: 1

Service time source: MATLAB action

Service time action:

```
%Call Simulink function 4 to obtain cargo tranfer rate for current
port:
CargoTransferRate=h(entity.To);

%Call simulink function 1 to obtain number of TEUs to be handled in
current port (unloaded+loaded):
AmountHandledPort=Amount(entity.To);

%Calculate time used for loading/unloading and mooring/unmooring
dt=AmountHandledPort/CargoTransferRate+1;
```

Entity server 13:

Tab: Main

Capacity: 1

Service time source: Dialog

Service time value: 0

Tab: Event actions

Event action:

Entry

Entry action:

```
%Make sure that Excel document is only accessed by xlsread function
first time the entity arrives in this block. Save to persistent
matrix:
persistent c fleet
coder.extrinsic('xlsread')

if isempty(c)
    fleet=zeros(36,13);
    fleet=xlsread('Fleet');
    c=1;
end

%Update previous visited port:
```

```
entity.From=entity.To;
%Update port call number:
entity.PortCall=entity.PortCall+1;

%Read next port of visit from fleet (Excel document saved in
persistent matrix)
entity.To=fleet(9+entity.PortCall,2+3*(entity.Period-1));

%If next "port" is transshipment meeting point, update attribute
Switch to 1.
%If not, update attribute Switch to 2
if entity.To==1;
    entity.Switch=1;
else
    entity.Switch=2;
end
%(Attribute Switch is used to guide the vessel in entity output
switch 1)
```

Simulink Function 1:

```
function AmountHandledPort = fcn(To)
%#codegen
global D
%Amount handled in port is equal to cargo on daughter vessel with
current
%port as destination times two. (assumed perfect balance between
ingoing
%and outgoing --> same number loaded as unloaded.
AmountHandledPort=D(To)*2;
```

Simulink Function 2:

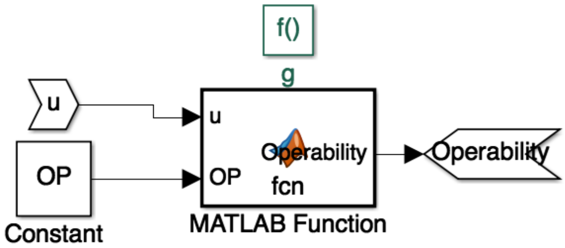


Figure 21:C. Layout Simulink function 2. OP from constant block (retrieved from workspace).

```
function Operability = fcn(u, OP)
%#codegen

%retrieve operability from constant block OP
Operability=OP;
```

Simulink Function 3:

```
function Speed = fcn(u, V)
%#codegen

%Retrieve speed from constant block V:
Speed=V;
```

Simulink Function 4:

```
function CargoTransferRate = fcn(To)
%#codegen

%Saving Excel document in persistent matrix. Also seed is persistent
and
%updated between each function call (for reproducability)
persistent seed fleet
coder.extrinsic('makedist','rng','xlsread')
if isempty(seed)
    seed=100;
    fleet=zeros(36,13);
    fleet=xlsread('Fleet');
end

%Parameters for triangular distribution from persistent matrix
(Excel):
A=fleet(32+To,3);
B=fleet(32+To,4);
C=fleet(32+To,5);

%Probability distribution (triangular):
pd=makedist('triangular',A,B,C);

%For reproducability:
rng(seed)

%Need to initialize size of CargoTransferRate as it cannot be an
mxArray
CargoTransferRate=zeros(1);
%Generate cargo transfer rate based on triangular distribution:
CargoTransferRate =random(pd);

%Update seed for next function call:
seed=seed+1;
```

Simulink Function 5:

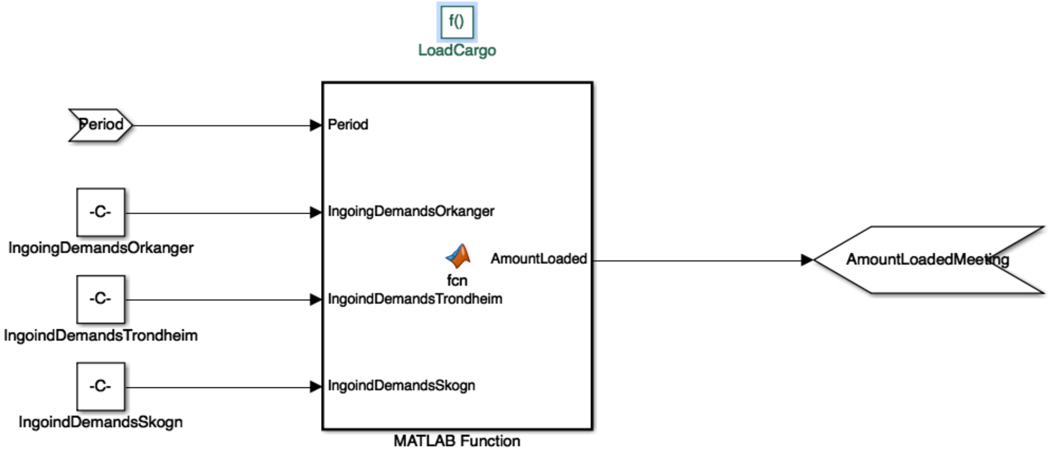


Figure 22: C. Layout Simulink function 5. Demands as constant blocks (retrieved from workspace).

```
function AmountLoaded =
fcn(Period,IngoingDemandsOrkanger,IngoindDemandsTrondheim,IngoindDem
andsSkogn)
%#codegen
%Declare meeting number and Excel document Fleet as persistent:
persistent meeting fleet
%Declare D as global variable
global D

coder.extrinsic('xlsread')

%Make sure that xlsread function is only run on first function call:
if isempty(meeting)
    meeting=1;
    fleet=zeros(36,13);
    fleet=xlsread('Fleet');
end

%Read which ports to visit in current period:
PortsToVisit=fleet(9:18,2+3*(Period-1));
VisitOrk=length(nonzeros(PortsToVisit==2));
VisitTrondheim=length(nonzeros(PortsToVisit==3));
VisitSkogn=length(nonzeros(PortsToVisit==4));

%Load cargo. Keep track of destination for cargo by loading into
global variable D which is
%vector with length 4. Accessing element in demands corresponding to
the meeting number:
D(2)=IngoingDemandsOrkanger(meeting)*VisitOrk;
D(3)=IngoindDemandsTrondheim(meeting)*VisitTrondheim;
D(4)=IngoindDemandsSkogn(meeting)*VisitSkogn;

%Total amount loaded. Send back to server:
AmountLoaded=sum(D);

%Update meeting number for next function call (new cargo amounts):
meeting=meeting+1;
```

Simulink Function 6:

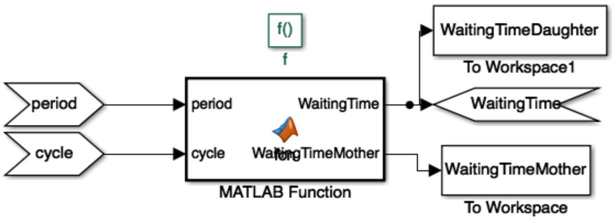


Figure 23: C. Layout Simulink function 6. WaitingTimeDaughter and WaitingTimeMother logged to workspace.

```
%Waiting time is calculated in this simulink function because the
global
%variable T (time) can not be accessed from a server.

%ALSO LOGS DAUGHTER VESSEL ARRIVAL TIME AT MEETING POINT TO
WORKSPACE

function [WaitingTime,WaitingTimeMother] = fcn(period,cycle)
```

```
global T
persistent c fleet
coder.extrinsic('xlsread')

if isempty(c) %This if statement makes sure that the Excel file is
only read the first time the function is called, and stored for
future calls. Imrproves the simulation time.
    fleet=zeros(36,13); %Initiate matrix
    fleet=xlsread('Fleet'); %Read Excel sheet into the matrix
    c=1; %Update c to 1. Prevents this section from being run again.
end

cycletime=fleet(1,7)*24;

time=T-cycle*cycletime;

MotherArrivalTime=fleet(6,3+3*(period-1));

%daughter vessel arrives before mothervessel:
if time <= MotherArrivalTime
    WaitingTime=MotherArrivalTime-time;
    WaitingTimeMother=0;

else %daughter vessel arrives after mother vessel
    WaitingTime=0;
    WaitingTimeMother=time-MotherArrivalTime;
end
```

Simulink Function 7:

```
%Calculate time until port opens:
function WaitToOpen = fcn
%#codegen
%Declare T as global variable to access it here
global T

%Declare persistent to make sure that xlsread function is only used
on
%first function call. Improves sim time
persistent fleet OpenHour ClosingHour
coder.extrinsic('xlsread')

if isempty(fleet)
    fleet=zeros(36,13); %Initiate size of matrix
    fleet=xlsread('Fleet'); %Read Excel document into matrix
    OpenHour=fleet(1,11); %Time that port opens
    ClosingHour=fleet(2,11); %Time that port closes
end

day=floor(T/24); %day number (0 corresponds to monday)
h=T-day*24; %hour number

if h < OpenHour %Arrives after midnight and before port opening hour
    WaitToOpen=OpenHour-h;
elseif h > ClosingHour %Arrives after closing hour and before
midnight
    WaitToOpen=24-h+OpenHour;
else % (   Openhour<=h<=ClosingHour   ) %Arrives when port is open
```

```
    WaitToOpen=0;
end
```

D. MATLAB script for running simulation model and processing output data

```
%Run simulation model from script:
%Clear workspace and close figures
clear all
close all
clc

%First open and run script to generate transportation demands
"Cargo" used
%as input for simulation model:
open('Cargo')
run('Cargo')

%simplify name of simulation model:
model = 'SimulationModel';
%Open simulation model:
open_system(model);

%Set simulation time
set_param(model, 'StopTime','336*200')

% RUN SIMULATION MODEL:
i=1; %Initialize row number for logging results
j=1; %Initialize colomn number for logging results

% RUN MODEL FOR DIFFERENT SPEEDS WHILE OPERABILITY OF SHIP-TO-SHIP
CARGO
% TRANSFER OPERATION IS CONSTANT. UPDATE OPERABILITY AND REPEAT:
%Set lower limit, step-length, and upper limit for evaluated
operability
for OP=0.8:0.1:1
    %Set lower limit, step-length, and upper limit for evaluated
speeds_
    for V=2.5:0.2:3.5
        sim(model);
        % Log waiting time for mother and daughter for each
simulation
        % (mean of all periods):
        WaitingTimeD(i,j)=mean(WaitingTimeDaughter.Data(5:end));
        Speed(j)=V;
        WaitingTimeM(i,j)=mean(WaitingTimeMother.Data(5:end));
        %(5:end) excludes four first meetings (cycle 1). Introduced
to
        %avoid start-up effects.
        %Share of meetings on time:
        Reliability(i,j)=length(nonzeros(WaitingTimeDaughter.Data))/
length(WaitingTimeDaughter.Data);

        %Separate each period when analyzing the results:
        h=1;
        %NB! Start on k=5 to exclude 4 first meeting (cycle 1).
Introduced
        %to avoid start-up effects.
        %NB! excluding last four meetings as daughter vessel not
always
        %finish all periods for low speeds.
        for k=5:4:length(WaitingTimeDaughter.Data) -4
            %Time spent waiting for mother vessel in each period:
            WaitingTimesDaughterP1(h)=WaitingTimeDaughter.Data(k);
            WaitingTimesDaughterP2(h)=WaitingTimeDaughter.Data(k+1);
            WaitingTimesDaughterP3(h)=WaitingTimeDaughter.Data(k+2);
            WaitingTimesDaughterP4(h)=WaitingTimeDaughter.Data(k+3);


            h=h+1;
        end
        %Average time spent waiting for mother vessel for each
period:
        WaitingTimeDP1(i,j)=mean(WaitingTimesDaughterP1);
        WaitingTimeDP2(i,j)=mean(WaitingTimesDaughterP2);
        WaitingTimeDP3(i,j)=mean(WaitingTimesDaughterP3);
        WaitingTimeDP4(i,j)=mean(WaitingTimesDaughterP4);

        %Minimum time spent waiting for daughter vessel for each
period:
        Min_WaitingTimeDP1(i,j)=min(WaitingTimesDaughterP1);
        Min_WaitingTimeDP2(i,j)=min(WaitingTimesDaughterP2);
        Min_WaitingTimeDP3(i,j)=min(WaitingTimesDaughterP3);
        Min_WaitingTimeDP4(i,j)=min(WaitingTimesDaughterP4);

        %Maximum time spent waiting for daughter vessel for each
period:
        Max_WaitingTimeDP1(i,j)=max(WaitingTimesDaughterP1);
        Max_WaitingTimeDP2(i,j)=max(WaitingTimesDaughterP2);
        Max_WaitingTimeDP3(i,j)=max(WaitingTimesDaughterP3);
        Max_WaitingTimeDP4(i,j)=max(WaitingTimesDaughterP4);


        %Share of meetings that daughter vessel arrives before
mother
        %vessel for each period:
        ReliabilityP1(i,j)=length(nonzeros(WaitingTimesDaughterP1))/
length(WaitingTimesDaughterP1);
        ReliabilityP2(i,j)=length(nonzeros(WaitingTimesDaughterP2))/
length(WaitingTimesDaughterP2);
        ReliabilityP3(i,j)=length(nonzeros(WaitingTimesDaughterP3))/
length(WaitingTimesDaughterP3);
        ReliabilityP4(i,j)=length(nonzeros(WaitingTimesDaughterP4))/
length(WaitingTimesDaughterP4);

        j=j+1;
    end
    j=1;
    i=i+1;
end

%RESULTS AVERAGE OF ALL PERIODS:
%Plot share of meetings daughter vessel arrives on time for all
evaluated
%speeds and operabilities of ship-to-ship cargo transfer operation:
```

```
figure;
marker=['^','o','X'];
for i=1:1:3;
p=plot(Speed(1,:),Reliability(i,:));
p.Color='k';
p.LineWidth=1;
p.Marker=marker(i);
xlabel('Speed [knots]')
ylabel('Share of meetings on time')
axis([-inf inf 0.5 1])
hold on
end
legend('OP=0.8','OP=0.9','OP=1')

hold off

%RESULTATER FROM EACH PERIOD (SEPARATED):
%Plot average time daughter vessel spend waiting for mother vessel
for each period:
figure;
plot(Speed(1,:),WaitingTimeDP1(1,:),Speed(1,:),WaitingTimeDP1(2,:),S
peed(1,:),WaitingTimeDP1(3,:),'LineWidth',1)
legend('OP=0.8','OP=0.9','OP=1')
hold off
title('P1')
hold off

figure;
plot(Speed(1,:),WaitingTimeDP2(1,:),Speed(1,:),WaitingTimeDP2(2,:),S
peed(1,:),WaitingTimeDP2(3,:),'LineWidth',1)
legend('OP=0.8','OP=0.9','OP=1')
hold off
title('P2')
hold off

figure;
plot(Speed(1,:),WaitingTimeDP3(1,:),Speed(1,:),WaitingTimeDP3(2,:),S
peed(1,:),WaitingTimeDP3(3,:),'LineWidth',1)
legend('OP=0.8','OP=0.9','OP=1')
hold off
title('P3')
hold off

figure;
plot(Speed(1,:),WaitingTimeDP4(1,:),Speed(1,:),WaitingTimeDP4(2,:),S
peed(1,:),WaitingTimeDP4(3,:),'LineWidth',1)
legend('OP=0.8','OP=0.9','OP=1')
hold off
title('P4')
hold off

%Plot share of meetings that daughter vessel arrives at meeting
point in
%time for all periods:
figure;
plot(Speed(1,:),ReliabilityP1(1,:),Speed(1,:),ReliabilityP1(2,:),Spe
ed(1,:),ReliabilityP1(3,:),'LineWidth',1)
```

```
legend('OP=0.8','OP=0.9','OP=1')
xlabel('Speed [knots]')
ylabel('Share of meetings on time')
title('P1')
axis([-inf inf 0 1])

figure;
plot(Speed(1,:),ReliabilityP2(1,:),Speed(1,:),ReliabilityP2(2,:),Spe
ed(1,:),ReliabilityP2(3,:),'LineWidth',1)
legend('OP=0.8','OP=0.9','OP=1')
xlabel('Speed [knots]')
ylabel('Share of meetings on time')
title('P2')
axis([-inf inf 0 1])

figure;
plot(Speed(1,:),ReliabilityP3(1,:),Speed(1,:),ReliabilityP3(2,:),Spe
ed(1,:),ReliabilityP3(3,:),'LineWidth',1)
legend('OP=0.8','OP=0.9','OP=1')
xlabel('Speed [knots]')
ylabel('Share of meetings on time')
title('P3')
axis([-inf inf 0 1])

figure;
plot(Speed(1,:),ReliabilityP4(1,:),Speed(1,:),ReliabilityP4(2,:),Spe
ed(1,:),ReliabilityP4(3,:),'LineWidth',1)
legend('OP=0.8','OP=0.9','OP=1')
xlabel('Speed [knots]')
ylabel('Share of meetings on time')
title('P4')
axis([-inf inf 0 1])
```

Input Excel documents for simulation model

Filename: Fleet

Ports:

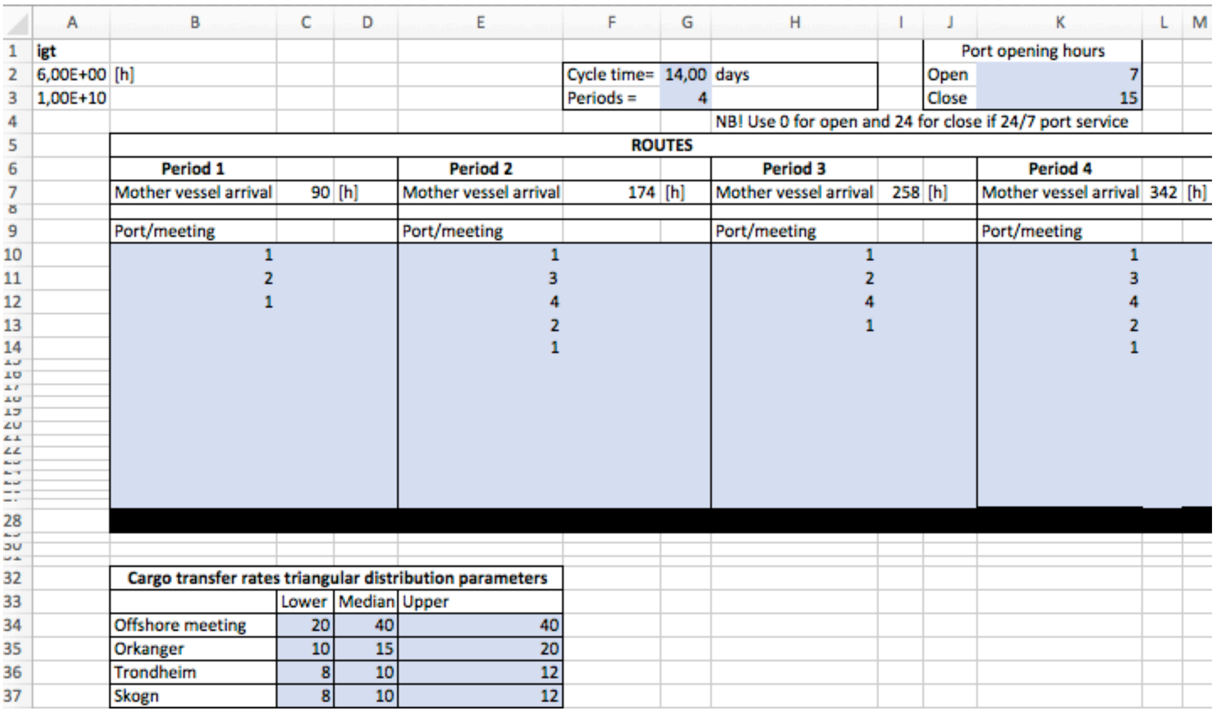
1: Transshipment meeting point

2: Orkanger

3: Trondheim

4: Skogn

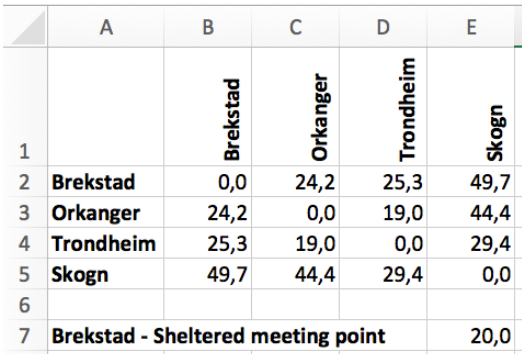
Configuration shown below corresponds to case B1, stochastic scenario



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	igt									Port opening hours			
2	6,00E+00	[h]				Cycle time=	14,00	days		Open	7		
3	1,00E+10					Periods =	4			Close	15		
4								NB! Use 0 for open and 24 for close if 24/7 port service					
5						ROUTES							
6		Period 1			Period 2			Period 3			Period 4		
7		Mother vessel arrival	90	[h]	Mother vessel arrival	174	[h]	Mother vessel arrival	258	[h]	Mother vessel arrival	342	[h]
8													
9		Port/meeting			Port/meeting			Port/meeting			Port/meeting		
10		1			1			1			1		
11		2			3			2			3		
12		1			4			4			4		
13					2			1			2		
14					1						1		
28													
32		Cargo transfer rates triangular distribution parameters											
33			Lower	Median	Upper								
34		Offshore meeting	20	40	40								
35		Orkanger	10	15	20								
36		Trondheim	8	10	12								
37		Skogn	8	10	12								

Figure 24: D. Input Excel file for simulation model. Filename: Fleet.

Filename: Distances



	A	B	C	D	E
1		Brekstad	Orkanger	Trondheim	Skogn
2	Brekstad	0,0	24,2	25,3	49,7
3	Orkanger	24,2	0,0	19,0	44,4
4	Trondheim	25,3	19,0	0,0	29,4
5	Skogn	49,7	44,4	29,4	0,0
6					
7	Brekstad - Sheltered meeting point				20,0

Figure 25: C. Input Excel file for simulation model. Filename: Distances.

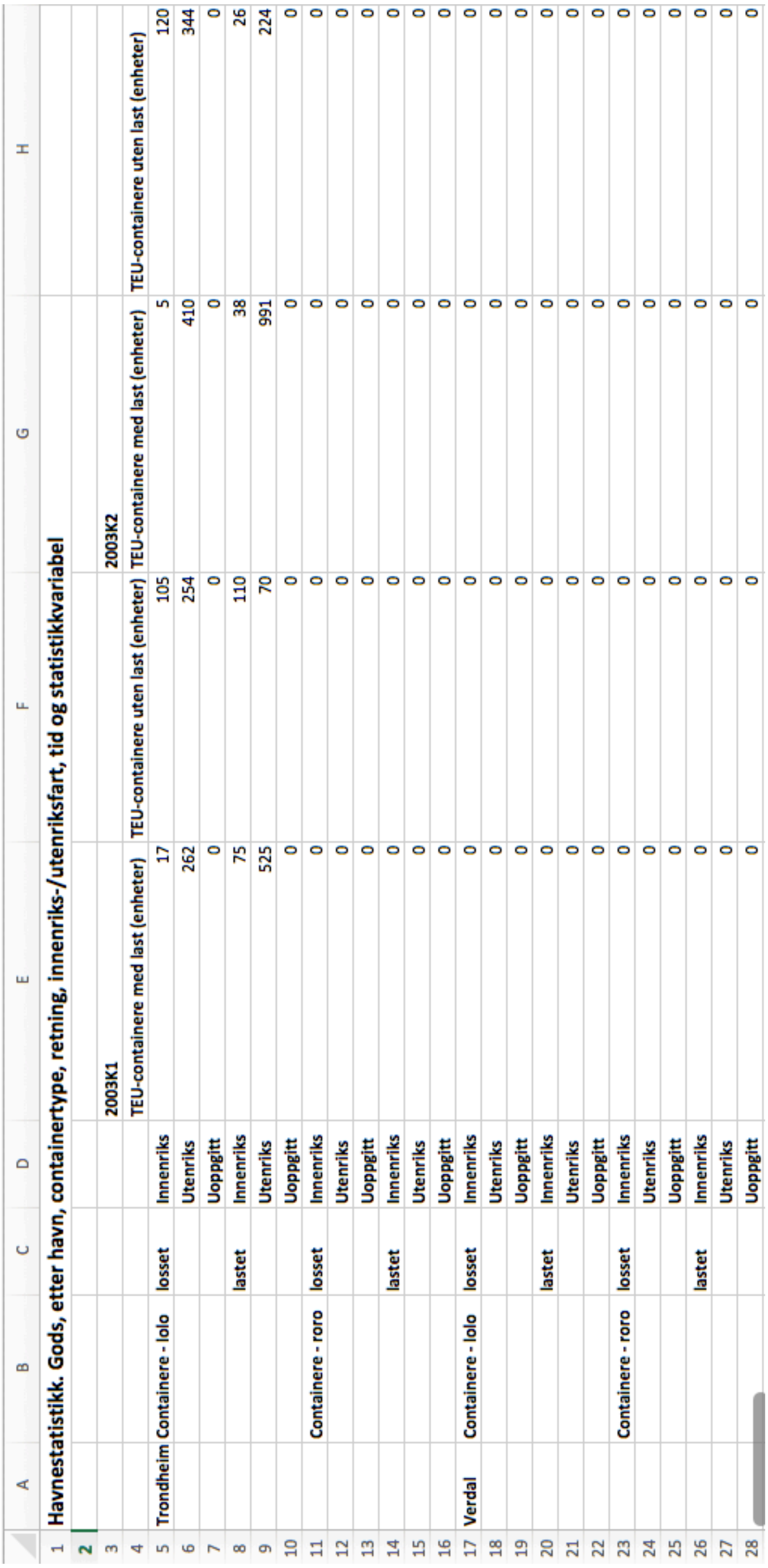
E. MATLAB script for generating transportation demands used as input in simulation model

```
%Generate transportation demands

%SET PARAMETERS (EXPECTED DEMAND GROWTH AND DEMAND SHARE BETWEEN
PORTS)
DemandGrowth=1.1;
Share=[0.80, 0.05, 0.15];
MarketShare=1;
nPeriods=4;
nVisits=[4 2 3];

%% SORT DATA FROM EXCEL SHEET:
Data=xlsread('20173171116118512200947Havnest5.xlsx');

Trondheim=Data(1:12,:);
Verdal=Data(13:24,:);

Total=Trondheim+Verdal;
lolo_Total=Total(1:6,:);
roro_Total=Total(7:12,:); %0 roro.

unloaded_AllDir=sum(lolo_Total(1:3,:));
loaded_AllDir=sum(lolo_Total(4:6,:));

c=1;
for i=1:2:length(Total)
    Filled_unloaded_AllDir(c)=unloaded_AllDir(i);
    Filled_loaded_AllDir(c)=loaded_AllDir(i);
    c=c+1;
end

c=1;
for i=2:2:length(Total)
    Empty_unloaded_AllDir(c)=unloaded_AllDir(i);
    Empty_loaded_AllDir(c)=loaded_AllDir(i);
    c=c+1;
end

Total_unloaded_AllDir=Filled_unloaded_AllDir+Empty_unloaded_AllDi
r;
Total_loaded_AllDir=Filled_loaded_AllDir+Empty_loaded_AllDir;

TotalContainerMoves_each_way=(Total_unloaded_AllDir+Total_loaded_
AllDir)/2;
```

```
%% Generate random realizations of container demands based on
uniform distribution, +-10% from mean:

%Set random number seed. (For reproducability)
rng(1234);
simValues=randi([round(mean(TotalContainerMoves_each_way)*(1-0.2))
round(mean(TotalContainerMoves_each_way)*(1+0.2))],10000,1);

%Plot histogram of simulated demands
figure;
histogram(simValues,'Normalization','pdf');
title('Simulated demands')

PeriodicDemands=simValues*DemandGrowth*MarketShare/(3*4*2);
%3 months in a quarter, 4 weeks in a month, 2 periods in a week

%% Values used as input in simulation model:
IngoingDemandsOrkanger=round(PeriodicDemands*Share(1)*(nPeriods/n
Visits(1)));
IngoingDemandsTrondheim=round(PeriodicDemands*Share(2)*(nPeriods/
nVisits(2)));
IngoingDemandsSkogn=round(PeriodicDemands*Share(3)*(nPeriods/nVis
its(3)));

%Alternatively for deterministic model run. Overwrite demand
vectors.
%    IngoingDemandsOrkanger(1:end)=round(mean(TotalContainerMoves_
each_way)*Share(1)*MarketShare*DemandGrowth/(3*4*2)*nPeriods/nVis
its(1));
%    IngoingDemandsTrondheim(1:end)=round(mean(TotalContainerMoves
_each_way)*Share(2)*MarketShare*DemandGrowth/(3*4*2)*nPeriods/nVi
sits(2));
%    IngoingDemandsSkogn(1:end)=round(mean(TotalContainerMoves_eac
h_way)*Share(3)*MarketShare*DemandGrowth/(3*4*2)*nPeriods/nVisits
(3));


%% Expected build-up of containers per period
PerPeriodIngoingOrk=round(mean(TotalContainerMoves_each_way)*Shar
e(1)*MarketShare*DemandGrowth/(3*4*2));
PerPeriodIngoingTrh=round(mean(TotalContainerMoves_each_way)*Shar
e(2)*MarketShare*DemandGrowth/(3*4*2));
PerPeriodIngoingSkogn=round(mean(TotalContainerMoves_each_way)*Sh
are(3)*MarketShare*DemandGrowth/(3*4*2));
```

Input Excel document from SSB:

Filename: 20173171116118512200947Havnest5



	A	B	C	D	E	F	G	H
1	Havnestatistikk. Gods, etter havn, containertype, retning, innenriks-/utenriksfart, tid og statistikkvariabel							
2								
3					2003K1		2003K2	
4					TEU-containere med last (enheter)	TEU-containere uten last (enheter)	TEU-containere med last (enheter)	TEU-containere uten last (enheter)
5	Trondheim	Containere - lolo	losset	Innenriks	17	105	5	120
6				Utenriks	262	254	410	344
7				Uoppgitt	0	0	0	0
8			lastet	Innenriks	75	110	38	26
9				Utenriks	525	70	991	224
10				Uoppgitt	0	0	0	0
11		Containere - roro	losset	Innenriks	0	0	0	0
12				Utenriks	0	0	0	0
13				Uoppgitt	0	0	0	0
14			lastet	Innenriks	0	0	0	0
15				Utenriks	0	0	0	0
16				Uoppgitt	0	0	0	0
17	Verdal	Containere - lolo	losset	Innenriks	0	0	0	0
18				Utenriks	0	0	0	0
19				Uoppgitt	0	0	0	0
20			lastet	Innenriks	0	0	0	0
21				Utenriks	0	0	0	0
22				Uoppgitt	0	0	0	0
23		Containere - roro	losset	Innenriks	0	0	0	0
24				Utenriks	0	0	0	0
25				Uoppgitt	0	0	0	0
26			lastet	Innenriks	0	0	0	0
27				Utenriks	0	0	0	0
28				Uoppgitt	0	0	0	0

Figure 26: Input Excel file for calculating transportation demand. First 8 of the 116 columns shown.

F. MATLAB script for calculating operability of cargo transfer operation

```
%Clear workspace
clear all
close all
clc

%Load wave data
load LensvikAndBrekstad

%Plot time series of significant wave height at Brekstad:
figure;
plot(Brekstad.Significant_wave_height);
xlabel('Time[h]')
ylabel('Hs[m]')


%% Set interval and step-length of operational criterion and
reference period to be analyzed:
%Operational criterin (Hs[m]):
OPwf=[0.5:0.25:1.5];
%Reference period [h]
Tr=[0:0.1:15];

%% Cumulative distribution waveheight (WEIBUL)
%Choose wave data to analyze:
wave=Brekstad.Significant_wave_height;

%Find all measurements with Hs=0m and change to Hs=0.01m:
%(wblfit requires only positive non-zero values.
%Will not affect results as all analyzed operational criterions
are > than
%0.01m)

m=length(wave);
for a=1:m
    if wave(a)==0
        wave(a)=0.01;
    end
end

%Estimate weibull parameters for waveheights:
param=wblfit(wave);

x=(0:0.001:1.5);
%Probability density function f:
```

```
f1=param(2)/param(1)*(x/param(1)).^(param(2)-1).*exp(-
(x/param(1)).^param(2));  %https://se.mathworks.com/help/stats/wbl
fit.html

%Plot fitted probability density function together with histogram
of forecasted significant wave
%heights:
figure;
plot(x,f1)
xlabel('Hs')
ylabel('f(Hs')
hold on
histogram(wave,'Normalization','pdf');
title 'BREKSTAD - Weibull PDF fitted to normalized histogram'

%Cumulative distribution function F:
F1=1-exp(-(x/param(1)).^param(2));

%Plot fitted cumulative probability distribution function together
with empirical
%cumulative probability distribution of forecasted significant wave:
figure;
c=cdfplot(wave);
c.Color='k';
xlabel('Hs[m]');
ylabel('F(Hs)');

grid off
hold on

plot(x,F1,'k','LineWidth',1)
title 'Empirical vs theoretical CDF'

hold off


%% Calcuate Probability that Hs is less than the evaluated OPwfs:
for i=1:length(OPwf)
    F_val(i)=wblcdf(OPwf(i),param(1),param(2));
end

%% Length of calms
figure;
plot(wave,'k')
xlabel('time[h]')
ylabel('Hs[m]')
color('blk')

for u=1:length(OPwf) %Calculate for all operational criterions
    i=1; %Hour in calm
    j=1; %Hour number
    k=1; % Calm number

for j = 1:length(wave) %Check wave height for all elements in the
time series
    if wave(j)<=OPwf(u) %Check if wave height at the time is lower
than operational criterion
        CalmVector(i)=j; %If yes, the element is part of a calm
        i=i+1;
        Lengths(k)=length(CalmVector); %The length of the current
calm is updated
    else
         CalmVector=0; %If no, the calm is completed
        k=k+1;        %Checking next calm
        i=1;          %Start on first hour in the calm
    end
end

%Lengths of all calms from current operational criterion:
LengthOfCalms_u=nonzeros(Lengths);

Lengths=0 % RESET for next loop (next operational criterion)

%Estimate weibull parameters for length of calms for current
operational
%criterion:
param_calm=wblfit(LengthOfCalms_u);

x_calm=(0:0.10:1000);

%Probability density function f:
f1_calm=param_calm(2)/param_calm(1)*(x_calm/param_calm(1)).^(para
m_calm(2)-1).*exp(-(x_calm/param_calm(1)).^param_calm(2));

%Cumulative distribution function F:
F1_calm=1-exp(-(x_calm/param_calm(1)).^param_calm(2));

%Plot empirical cumulative distribution function vs estimated
weibull
%function:
figure;
cc=cdfplot(LengthOfCalms_u);
cc.Color='k';

hold on
plot(x_calm,F1_calm,'k','LineWidth',1)
title(['Hs=' num2str(OPwf(u))])
xlabel('Length of calm[h]');
ylabel('F(Length of calm)');
grid off
```

```
axis([0 1000 0 1])


%% Calculate availability of operation:
for v=1:length(Tr)

                                                               probCalm(u,v)=1-
wblcdf(Tr(v),param_calm(1),param_calm(2)); %[wave height,referance
period]

    ProbWork(u,v)=F_val(u)*probCalm(u,v);

end

end

% Plot availability of the operation as function of referance period
for
% each evaluated operational criterion:
%(2016aplot)
figure;
for i=1:length(OPwf)
    plot(Tr,100*ProbWork(i,:),'linewidth',1)
    xlabel('Reference period [h]')
    ylabel('Availability of operation [%]')
    axis([min(Tr) max(Tr) 0 100])
    set(gca,'xtick',[0:1:max(Tr)]);
    set(gca,'ytick',[0:10:100]);
    hold on
end
legend('OpLim=0.5m',  'OPLim=0.75m',  'OPLim=1m',  'OPLim=1.25m',
'OPLim=1.5m')

%PLOTTING WITH MARKERS ON ONLY SOME OF THE DATA POINTS. NB! THIS
PLOT IS
%ONLY APPLICABLE FOR MATLAB R2016b OR NEWER:
figure;
for i=1:length(OPwf)
    marker=['^','o','X','+','d'];
    p=plot(Tr,100*ProbWork(i,:));
    p.Color='k';
    p.LineWidth=1;
    p.Marker=marker(i);
    p.MarkerIndices=[40:40:length(Tr)-5];
    xlabel('Duration of operation [h]') % Reference period
    ylabel('Availability of operation [%]')
    axis([min(Tr) max(Tr) 0 100])
    set(gca,'xtick',[0:1:max(Tr)]);
    set(gca,'ytick',[0:10:100]);
    hold on
```

```
end
legend('OpLim=0.5m',  'OPLim=0.75m',  'OPLim=1m',  'OPLim=1.25m',
'OPLim=1.5m')
```