NTNU
Norwegian University of
Science and Technology

Evaluation of Autonomous Container Feeder Fleets in Different Contexts and Needs

Harald Bendik Ulvestad Salvesen

Marine Technology
Submission date: June 2018
Supervisor: Bjørn Egil Asbjørnslett, IMT

Norwegian University of Science and Technology
Department of Marine Technology

Preface

This thesis is the final part of my Master of Science (MSc) degree in Marine Technology with specialization in Marine Design and Logistics at the Norwegian University of Science and Technology. The work has been carried out during the spring of 2018, and the workload corresponds to 30 ECTS.

This thesis is a continuation of my project thesis written in the fall of 2017, with the topic "Fleet Evaluation Considering Different Contexts & Needs".

I would like to thank my supervisor, Professor Bjørn Egil Asbjørnslett, for advice and support during the fall semester of 2017 and spring semester of 2018. I want to extend my sincerest thanks to PhD Carl Fredrik Rehn for providing guidance, valuable advice and relevant literature throughout the final year of my MSc degree. Your support has been much appreciated. I would also like to express my gratitude to PhD Sigurd Solheim Pettersen for programming guidance in MATLAB and Xpress IVE.

Trondheim, June 27, 2018

Harald U. Salvesen

Harald Ulvestad Salvesen

Abstract

Unmanned autonomous ships are a popular subject in recent maritime research and innovation studies. Autonomy in shipping bears the potential to reduce the environmental impact, increase safety and reduce costs. So far, minimal attention has been devoted to the potential cost benefits of autonomous shipping. From a ship owner's perspective the main determinant of autonomous shipping rely on the economic viability. Ship acquisition involve a large capital investment, and the future operating context is highly uncertain. The maritime industry is characterized by uncertain exogenous factors that affect operational costs.

The overall objective of this thesis is to compare autonomous container feeder fleets to conventional feeder fleets in terms of costs. In order to investigate potential cost-benefits of autonomy, the required infrastructure and technical systems are identified. The costs changes that occur for autonomous vessels are estimated using approximation methods with a conventional container vessel as reference. Operational costs and voyage costs are determined by solving a multi-trip Vehicle Routing Problem for a specific regional Baltic container trade. Contextual uncertainties are considered using the Responsive Systems Comparison method together with Multi-Epoch Analysis.

The results from the analysis show that autonomous container fleets have lower costs than conventional container fleets in a variety of different contexts. In a base case scenario the cost of the optimal autonomous fleet is USD 1.39 million lower than for optimal the conventional fleet. However, the results from the analysis are related with a high degree of uncertainty. The uncertainties are primarily due to the conceptual state of autonomy in container shipping, and limited access to precise cost data for autonomous vessels. It turned out to be challenging to capture the difference between autonomous and conventional vessels using the Responsive Systems Comparison method. Eventually, the method acted more like a sensitivity analysis. However, in a future scenario when autonomous shipping is more developed, the overall approach can serve as a powerful tool for fleet evaluation.

Sammendrag

Ubemannende autonome skip er et populært tema i nyere maritim forskning og innovasjonsstudier. Autonomi i shipping har potensiale til å redusere miljøpåkjenninger, øke sikkerheten og redusere kostnader. Så langt har det ikke vært viet vesentlig oppmerksomhet til de potensielle kostnadsfordelene som autonomi kan innebære. Fra ståstedet til en skipsreder er økonomisk levedyktighet den mest avgjørende faktoren. Oppkjøp av skip innebærer store kapitalinvesteringer, hvor den fremtidige operative konteksen er meget usikker. Karakteristisk for den maritime industrien er usikre eksterne faktorer som påvirker operasjonelle kostnader.

Denne oppgavens overordnede mål er å sammenligne autonome container feeder-flåter med konvensjonelle feeder-flåter når det gjelder kostnader. For å kunne undersøke de potensielle kostnadsfordelene med hensyn til autonomi, må den nødvendige infrastrukturen og de tekniske systemene identifiseres. Kostnadsendringene som oppstår ved autonome skip estimeres ved å bruke approksimeringsmetoder med et kovensjonelt containerskip som referanse. Operasjonelle kostander og reisekostnader betemmes ved å løse et multi-trip Vehicle Routing Problem for en spesifikk regional containerhandel (Baltic). Kontekstuelle faktorer blir tatt hensyn til ved å bruke Responsive Systems Comparison-metoden sammen Multi-Epoch analyse.

Resultatene fra analysen viser at autonome containerflåter har lavere kostnader enn kovensjonelle containerflåter i en rekke ulike kontekster. Kostnadene til den optimale autonome containerflåten i et base case scenario er 1.39 millioner USD lavere enn den optimale konvensjonelle containerflåten. Derimot er resultatene av denne analysen forbundet med stor grad av usikkerhet. Usikkerheten skyldes primært den konseptuelle tilstanden innen autonomi i container-shipping, og begrenset tilgang til nøyaktige kostnadsdata for autonome skip. Det viste seg å være utfordrene å identifisere forskjellene mellom autonome og konvensjonelle skip i Responsive Systems Comparison-metoden. Metoden viste seg å fungere mer som en sensitivitetsanalyse. I midlertid kan den overordnede tilnærmingen fungere som et nyttig verktøy for flåteevaluering i et fremtidig scenario hvor autonome skip er mer utviklet.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Shipping accounts for nearly 80% of global trade by volume, which in 2016 reached about 10.6 billion tons of cargo (Brouer et al., 2016). Seaborne trade is certainly a global industry, and is connected to almost every international supply chain. The demand for seaborne cargo capacity is increasing as global population and standard of living rise (Christiansen et al., 2012).

The shipping industry has seen several innovative solutions to improve efficiency since the first cargoes were shipped by sea more than 5,000 years ago. The containerization of general cargo during the 1950s drastically decreased cargo handling time, which resulted in improved efficiency and cost reductions (Stopford, 2009). In more recent times, digital solutions and data analytics have made it possible to further improve shipping efficiency, such as optimized weather routing and condition monitoring. Operations research is common in the transportation industry to minimize costs and obtain competitive logistic solutions. Containerized cargo constitutes about 60% of all value shipped by sea, but in the last decades minimal attention has been devoted to operations research within containerized shipping compared to other transportation modes (Ronen et al., 2004). However, in the past decade the research activity within maritime transportation planning problems is increasing (Christiansen et al., 2012).

Thorough planning may have a significant impact on shipping costs and operational efficiency. Seaborne transportation constitute about 2.7% of global CO_2 emissions, where 25% is accounted to container shipping alone (Løfstedt et al., 2010). Thus, proper planning also has the potential to reduce emissions.

Autonomous vehicles are emerging in transportation fields such as aviation, the automotive and train industry. Aircrafts have been semi-autonomous for several decades by the use of the autopilot system. Leading companies like Google and Tesla have invested significant amounts in fully autonomous cars for the everyday consumer. In recent maritime research and innovation studies, unmanned and autonomous ships have become a popular research subject. According to (Burmeister et al., 2014 & Rodseth et al., 2012) the incentive behind autonomous vessels in the maritime sector is to contribute to all dimensions of sustainability; economic, ecological and social sustainability.

Although the number of accidents in the maritime transportation is low compared to other transportation modes, it is a necessity that autonomous vessels are at least as safe, or even safer, than today's manned vessels (Kretschmann et al., 2017). Several studies show that the majority (from 64% to 96%) of maritime accidents are due to human error (Sanquist, 1992, Blanding, 1987 and Rothblum, 2000). Autonomous vessels have the potential to improve navigational safety by replacing the officer of the watch (OOW) with automated onboard decision-making systems. Together with the onboard systems, the autonomous vessels are operated by a onshore nautical officer from a so-called Shore Control Centre (Burmeister et al., 2014). This may also attract seagoing professionals due to more attractive working conditions.

From an economic perspective the introduction of unmanned vessels has the potential to reduce operating costs by removing the majority of crew-related costs. The absence of onboard crew may also lead to new innovative ship designs without the need for superstructures to house life-support facilities. Losing the superstructure means less air resistance and also a reduction in weight, which in turn may result in reduced fuel consumption. According to Bertram (2016), so far minimum attention has been devoted to the potential cost benefits of autonomous vessels in the maritime transportation sector.

From a shipowner's perspective, a potential procurement of autonomous vessels involves a large capital investment. The shipping market is a highly uncertain market, characterized by unpredictable and exogenous factors affecting costs. It's reasonable to assume that in order for a ship owner to switch to autonomous vessels, the main determinant factor would be potential cost benefits relative to conventional vessels. As the world is constantly changing, it is crucial that new ships design are able to add value throughout their lifetime. Ships designed using traditional design methods often rely on a static operating context. However, there is no guarantee that the ship will stay optimal if the future operating context change (Ross et al., 2008b).

1.2 Objectives

The overall objective of this thesis is to compare conventional container feeder fleets to autonomous feeder fleets. There are several ways to conduct a comparison study, but from a potential shipowner's point of view the most interesting measure is cost-comparison. This thesis aims to answer the following research question:

> *Can the container shipping industry benefit of autonomous container feeder fleets?*

In order to investigate potential benefits of autonomous container vessels, it is necessary to conduct a thorough assessment of the required frameworks for autonomous operation. In addition, the cost changes that may occur due to innovative autonomous ship designs are identified. To handle the uncertainties in the shipping market and ensure value robust designs, a second research question is defined:

> *What configuration of autonomous container feeder fleets minimizes cost and performs best in different contexts and needs?*

1.3 Scope and Limitations

In this thesis the comparison study between autonomous and conventional vessels is restricted to container feeder networks in liner shipping. The analysis deals with technical and economical issues related to autonomous vessels, where legal and safety issues are not considered. The possibility to retrofit conventional feeder vessels into autonomous vessels is neglected in this analysis, and it's assumed that a carrier would prefer to acquire new autonomous vessels. It's plausible that a retrofitting process would be too extensive. The main focus of this thesis is seaborne transportation, thus intermodal logistics are not considered.

The main limitation in this thesis is related to the conceptual state of autonomous shipping, and the associated uncertainties connected to costs. The results of the cost assessment rely solely on the assumptions and estimates made, and the underlying cost data for similar container vessels. Another limitation is the availability of cost data, especially related to autonomous ships.

1.4 Structure of the Report

The rest of the thesis is structured as follows:

- **Chapter 2** identifies the considered differences between conventional and autonomous container vessels. The chapter starts out with a definition of an autonomous vessel, and the required infrastructure and technical systems are investigated. Furthermore, a conventional reference vessel is presented, and the cost differences that occur for an autonomous vessel are presented.
- **Chapter 3** presents the structured approach for the analysis. Relevant theory is explained to give the reader insight in the methodologies used. First, methods for solving Maritime Transportation Planning Problems (MTTP) are presented. Next, the focus shift towards the Responsive Systems Comparison (RSC) method, Tradespace Exploration and Epoch-Era Analysis (EEA).
- **Chapter 4** reviews relevant literature related to the approach and scope of this thesis. This involves similar research topics on routing and scheduling, the RSC approach and cost assessments of autonomous vessels.
- **Chapter 5** presents the solving algorithm for the routing problem using a pipeline flow chart. The mathematical optimization models used to solve the problem are presented and explained.
- **Chapter 6** presents a case study that involves a specific regional Baltic container trade. The RSC method is applied and the results of the analysis are presented.
- **Chapter 7** provides a discussion of the results and the methods used.
- **Chapter 8** presents the conclusion of the thesis, and recommendations for further work are proposed.

Chapter 2

Conventional vs. Autonomous

In this chapter the differences between conventional and autonomous container vessels are identified and discussed. The first part of this chapter defines an autonomous vessel, and presents existing research projects that investigate the commercial feasibility of autonomous vessels in the maritime transportation sector. Furthermore, the data used to estimate the costs of owning and operating a conventional container vessel is presented. The differences that occur when switching to autonomous operation are then discussed and structured.

2.1 What is an Autonomous Ship?

According to the European Waterborne Technology Platform Implementation Plan (Waterborne TP, 2011) for 2020, the development of autonomous ships is one of the key *exploitation outcomes* to strengthen Europe's maritime sector. Waterborne TP (2011) defines an autonomous ship as a vessel with:

> *Next generation modular control system and communications technology* [*that*] *will enable wireless monitoring and control functions both on and off board. These will include advanced decision support systems to provide a capability to operate ships remotely under semi or fully autonomous control.*

In order to bring the idea of autonomous ships to life, the research project *Maritime Unmanned Navigation through Intelligence in Networks* (MUNIN) was established by the European Commission. MUNIN investigates the feasibility of autonomous ships, and aims to develop required technology and business concepts (Kretschmann et al., 2017). As outlined in the definition of an autonomous ship, the vessel should be able to operate remotely under *semi* or *fully* autonomous control. The MUNIN project's core task is to develop and validate the required technology to achieve semi or fully autonomous control (Rodseth et al., 2012). According to the MUNIN project a ship is said to be autonomous if it is completely unmanned at least for parts of a particular voyage (MUNIN, 2016).

2.1.1 Autonomous Infrastructure

Figure 2.1 illustrates the transition from manned vessels to autonomous vessels where the resulting autonomous ship is a combination of a remote and an automated ship. On a conventional manned ship, information from radars, navigation systems (ECDIS) and visual observations are interpreted by the onboard Officer on Watch (OOW) to take navigational actions. On the other hand, a remote ship transmits data to a navigational operator ashore where the information is processed. Navigational instructions are then transmitted back to the ship to take actions. The next level is the automated ship, where all navigational decisions are made onboard by a computer without any remote control. Finally, as defined by the MUNIN project, an autonomous ship is a combination of a remote and automated ship, where the vessel is completely unmanned for parts of the voyage. Navigational decisions are partially made by onboard automated systems and by navigational operators that are capable of controlling the ship remotely ashore.

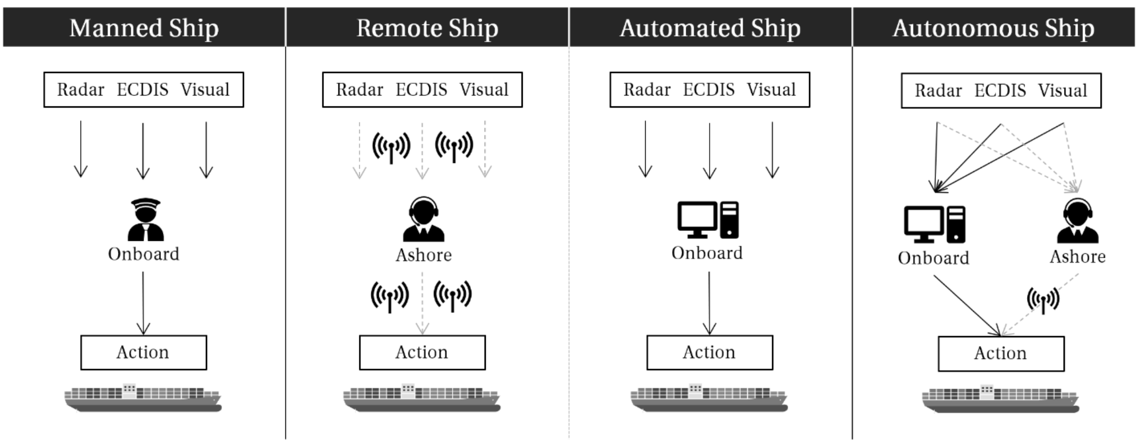

Figure 2.1: From manned to autonomous ships (Rodseth et al., 2012). Vector icons made by: `www.freepik.com`

In order to achieve safe unmanned and autonomous navigation it is necessary to implement newly developed systems on-board the vessel, and remote monitoring and control systems ashore (MUNIN, 2016). As proposed by the MUNIN project, autonomous operation requires voyage planning, navigation and collision avoidance systems that are constantly surveilled from a so-called Shore Control Centre (SCC) (MacKinnon et al., 2015). The monitoring and controlling tasks are conducted by a operator in the SCC, which is capable of operating up to six ships simultaneously (MUNIN, 2016). The SCC communicates with the autonomous ship by using available technology such as GSM, satellite, VHF, etc. From a socio-technical perspective the SCC can either be managed by officials such as the Vessel Traffic Services (VTS) and port authorities, or ship management companies to attract seagoing professionals looking for jobs ashore (MacKinnon et al., 2015). This conceptual overview, as proposed by MacKinnon et al. (2015), is illustrated in Figure 2.2.

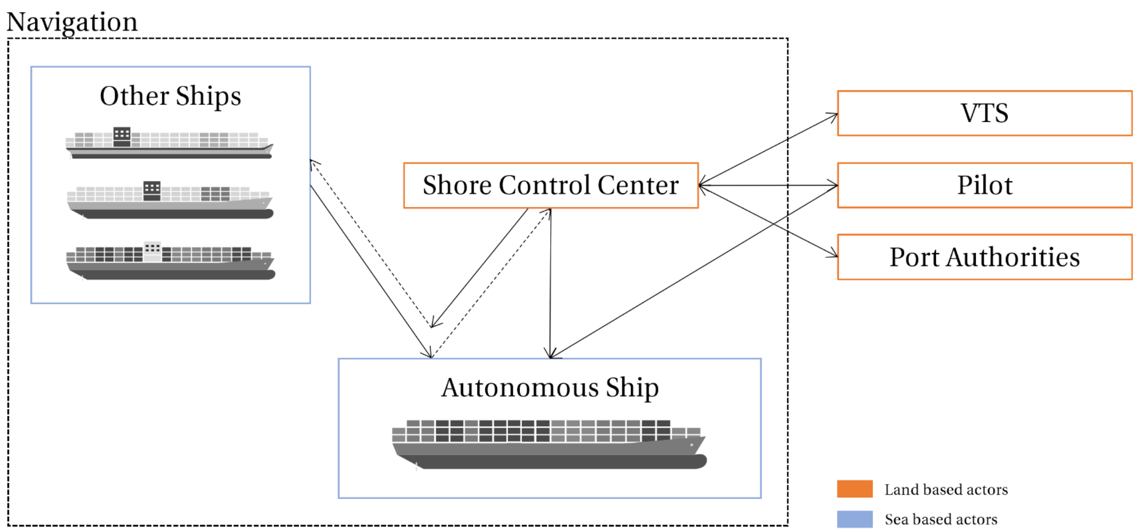

Figure 2.2: Socio-technical overview of autonomous navigation (adapted from MacKinnon et al., 2015). Vector icons made by: `www.freepik.com`

When necessary, it is possible to take direct control of the autonomous vessel from the SCC by using a Remote Maneuvering Support System (RMSS). The idea is that a team ashore directly operate the vessel from a replica of the ship's bridge, and the RMSS provide the team with situation awareness due to the actual distance between the SCC and the vessel.

An Advanced Sensor Module (ASM) on-board the autonomous vessel replace the eyes and ears of the OOW. The ASM is equipped with advanced cameras and radars to detect and classify objects, and can be seen as the vessel's perception unit (MUNIN, 2016). A predefined voyage plan uploaded from the SCC is interpreted by an Autonomous Navigation System (ANS) on-board the vessel. The ANS has some degrees of freedom and can change the route if necessary, e.g., if harsh weather conditions suddenly arise. Together with the ANS, an Autonomous Engine Monitoring and Control System (AEMCS) transmit engine and navigational data to the SCC so that operators ashore can quickly identify hazards and threats. The AEMCS is an extension to existing ship automation and control systems, where the goal is to add required digital interfaces to allow for autonomous operation of technical systems on-board (MUNIN, 2016).

These high-level systems and components are considered the most important ones in order to capture the cost difference between autonomous and conventional vessels. There are certainly more detailed technical systems and infrastructure needed in a real-world situation, but as autonomous shipping is still in its conceptual phase, it's difficult to include all of the necessary elements. The link between the key systems and infrastructure considered, as presented by the MUNIN project, are illustrated in Figure 2.3.

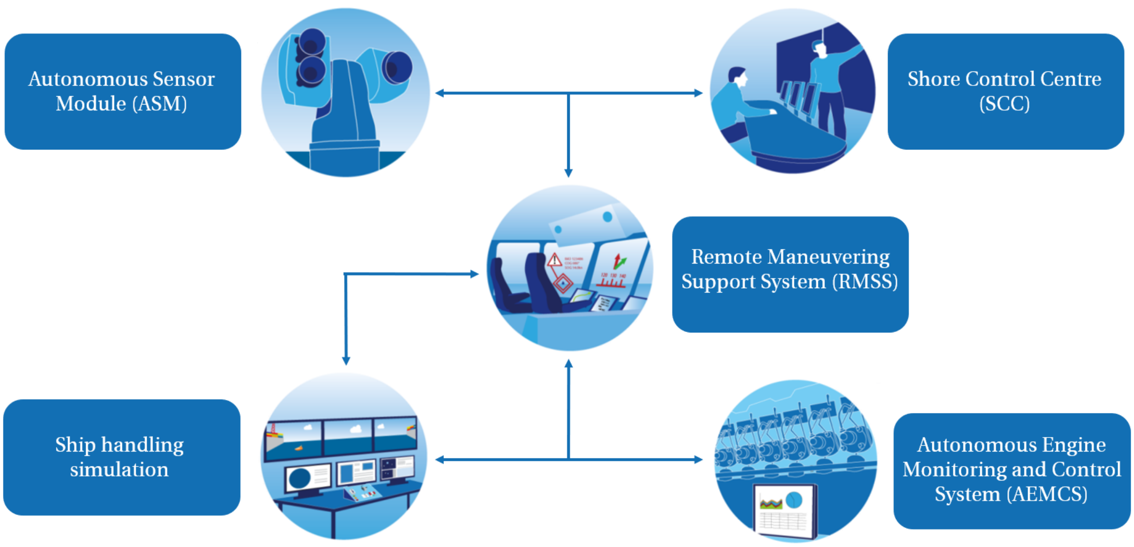

Figure 2.3: Overview of technical autonomous systems (reproduced from MUNIN, 2016).

2.1.2 Autonomous Ship Design

The introduction of unmanned and autonomous ships attracts innovative design solutions. Since the vessels are partially or fully unmanned, major parts of the superstructure, if not all of it, are no longer needed. Conceptual and real autonomous ship designs have recently emerged.

Yara Birkeland, the world's first electric driven autonomous container feeder, is ready for operation by 2020. The vessel has cargo capacity of 120 TEUs, and is capable of replacing around 40,000 truck deliveries annually between Larvik, Brevik and Herøya in Norway. Yara Birkeland is going to serve as a transportation "pendulum" between the three ports, and berthing, loading and discharging are fully autonomous processes using automatic mooring systems and electric cranes. Instead of using ballast tanks, Yara Birkeland will use battery packs as permanent ballast. As depicted in Figure 2.4, superstructure and crew facilities found on conventional vessels are completely eliminated from Yara Birkeland's design. Due to these design solutions the building process is promised to be less time consuming compared to conventional container vessels (Stensvold, 2017b). Unfortunately, production cost and operating cost data are not available at this point.

Figure 2.4: Yara Birkeland as designed by Marin Teknikk (Kongsberg, 2017).

SCCs deal with operational aspects including emergency and exception handling, condition monitoring, decision support and monitoring of the surroundings. Yara Birkeland is apparently ready for testing during 2018 with a captain and small crew placed in a container-based bridge module (Kongsberg, 2017). The bridge is detachable and will be removed when the vessel is ready for autonomous operation.

Rolls-Royce Marine has taken the concept of a module based autonomous container vessels to another level with their concept design called Electric Blue. The vessel is a 1,000 TEU container feeder with a unique flexible design that can easily adapt to changes in the market. Electric Blue's propulsion can be diesel-electric, LNG-electric or fully electric for specific routes or to meet future environmental regulations. This is made possible by introducing modular container components such as battery packs, fuel tanks, engine packs and even cabin packs (Stensvold, 2017a). This flexibility is a huge advantage in the volatile shipping market characterized by unpredictable exogenous factors such as environmental regulations and fluctuating fuel costs (Wilson, 2017). As depicted in Figure 2.5 the lean design with minimal superstructure makes more room for container cargo, and as for Yara Birkeland, the control bridge is housed in a container. Rolls-Royce Marine predicts that full autonomy is possible by 2035 (Wilson, 2017).

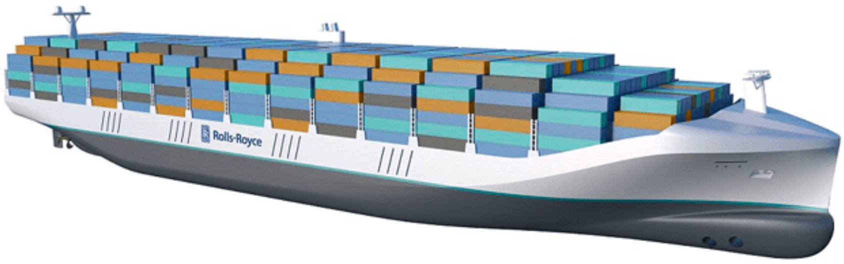

Figure 2.5: Initial concept design drawing of Electric Blue. Source: Rolls-Royce.

2.2 Classification of Costs

In this section, an overall cost structure for owning and operating a conventional container ship is presented. Further, a cost model for the conventional container vessel is described, and then the cost differences that occur when owning and operating an autonomous container vessel are discussed and structured.

The shipping industry has no internationally accepted standard for classifying costs, but on a parent level there are three main factors that affect the cost of running a ship (Stopford, 2009). The first factor involve the *state* of the vessel, namely its fuel consumption, required crew number, age and physical condition. The latter will affect the vessel's need for repairs and maintenance. Second, *external factors*, such as the price of equipment and bunkers, crew wages, repair and interest rates, are factors that are considered to be outside the shipowner's control. The last factor, *management*, depend on how well the ship is managed, including administrative overheads and operational efficiency (Stopford, 2009). The breakdown of factors that affect running costs for a ship is illustrated in Figure 2.6.

A merchant ship's ability to generate revenue rely on three main characteristics; cargo capacity, productivity and freight rates. Autonomous vessels need to compete with conventional vessels, so it is realistic to assume that same economical principals apply for autonomous vessels (Kretschmann et al., 2017). According to Stopford's (2009)

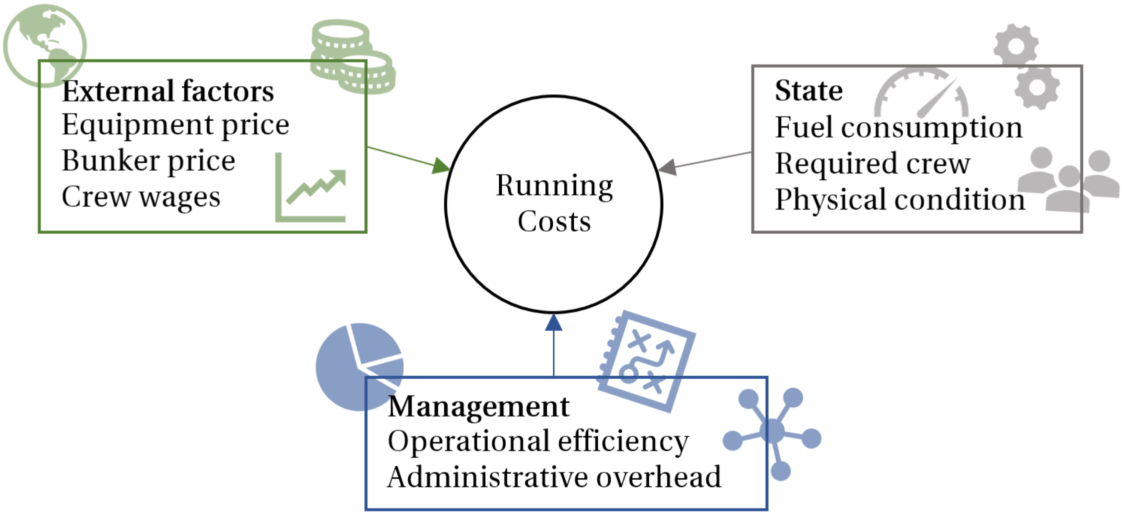

Figure 2.6: Breakdown of factors that affect the costs of running a ship (adapted from Stopford, 2009).

cash-flow model for merchant ships, the revenue generated by the vessel creates free cash flow after costs are deducted, as illustrated in Figure 2.7. The free cash flow is then used to pay tax and dividends, and generate profit.

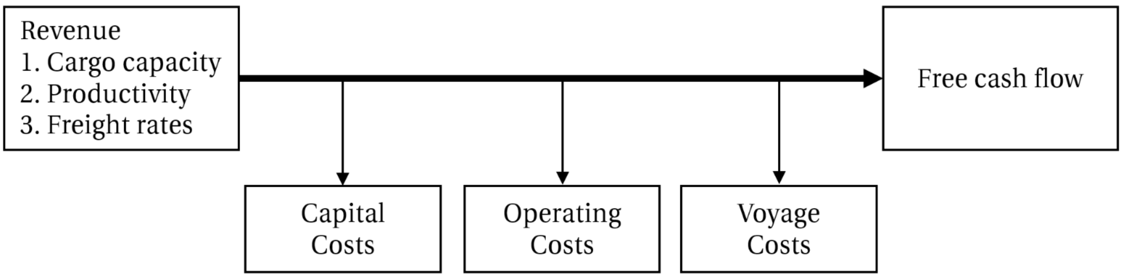

Figure 2.7: Adapted version of Stopford's shipping cash flow model (adapted from Stopford, 2009).

In thesis, three main cost categories are considered, and the following cost categories will be the basis for the cost models for the conventional and autonomous container vessel:

Operating Costs

Operating costs, commonly referred to as OPEX, are the day-to-day running costs of a vessel, such as crew, stores and maintenance. In other words, operating costs involve all costs needed to maintain the ship in operation, except fuel costs. On a general basis operating costs amount for about 14 % of total costs. Operating costs depend on the crew size and the nationality of the crew, maintenance policy, and age and insured value of the ship (Stopford, 2009). Periodic maintenance that involve dry docking is a major expense, and is considered as a fixed cost for the ship owner (Kretschmann et al., 2017).

Voyage Costs

Voyage costs, or VOYEX, are variable costs only associated with a particular voyage. The main elements are fuel costs, port call costs, tugging, canal charges and cargo-handling costs. The most important element is fuel cost as it accounts for about 45% of voyage costs (Stopford, 2009). Port charges are also considered a major expense, and may vary according to area, volume of cargo and vessel size.

Capital Costs

Capital costs, or CAPEX, are all expenses associated with ship acquisition, and related costs, such as interest (Kretschmann et al., 2017). Capital costs depend on how the vessel is financed, but apposed to operating costs and voyage costs, capital costs have no direct influence on the vessel's physical operation.

2.3 Cost Data for Conventional Container Vessels

The cost estimates for conventional container vessels are based on three separate data sources that serve as the input for the regression models presented later in chapter 6.1. The first data set, Table 2.1, is based on annual operational costs presented by Drewry (2010), and costs presented by Stopford (2009). The second data set, Table 2.2, provides cost data and power output for the main and auxiliary machinery. The data set consists of 1045 vessels with capacity ranging from 300 TEU to 1500 TEU.

Table 2.1: Selection of conventional container vessel data from Stopford (2009) & Drewry (2010).

Capacity [TEU]	Speed [kn]	Fuel consumption [t/h]	OPEX [USD/h]	CAPEX [millUSD]
660	18.3	1.17	146.0	-
1,200	18.3	1.75	193.5	25
1,216	22.0	2.66	154.7	-
2,468	21.0	4.07	199.0	-
2,600	20.9	3.29	237.8	48
3,752	21.5	5.83	265.6	-
4,300	23.8	6.13	250.0	67
5,364	25.0	6.88	297.0	-

Table 2.2: Selection of ship data from Sea Web. Source: www.maritime.ihs.com

Capacity [TEU]	Length [m]	Fuel Consumption [t/h]	Speed [kn]	dwt [t]	CAPEX [millUSD]	Main Engine [kW]	Aux. Engine [kW]
378	118.6	0.33	12.0	6,300	-	2,500	240
411	111.2	0.65	14.3	6,273	9.0	3,360	-
658	118.3	1.04	17.5	6,850	-	6,100	540
698	132.4	1.34	17.5	8,672	13.0	5,000	-
750	134.7	1.42	18.3	9,167	-	7,300	400
797	133.2	1.35	18.0	9,865	17.0	7,500	-
812	140.6	1.54	18.0	9,322	18.0	8,400	470
907	145.1	1.50	17.8	12,601	18.3	7,988	-
957	139.1	1.54	18.8	11,846	16.8	9,600	1,000
1,085	157.9	1.63	18.5	14,220	24.5	9,960	1,360
1,114	147.8	1.71	19.6	13,684	23.0	9,730	810
1,432	182.9	2.00	19.1	24,244	24.0	11,768	1,350

2.4 Cost Changes for Autonomous Container Vessels

In this section the considered cost changes that occur for an autonomous container vessel are presented and discussed. The cost estimates are structured according to operating costs, voyage costs and capital costs. The following estimation methods are primarily based on MUNIN's quantitative analysis of the concept of an autonomous vessel, where a panamax dry bulk carrier is used for reference (Kretschmann et al., 2015). The methods are adjusted according to the conventional container vessel used as a reference in this analysis.

The conventional container vessel considered in this thesis, hereafter referred to as the *reference vessel*, is a small container feeder with capacity of 800 TEU. Feeders are often categorized into three subcategories according to their capacity; small feeder (up to 1,000 TEU), feeder (1,000-2,000 TEU) and feedermax (2,000-3,000 TEU), but the term feeder can be used for all categories. The reference vessel is built by Damen Shipyard Group, and the vessel's main specifications are listed in Table 2.3. The main specifications given in the vessel's product sheet (Damen, 2017) are needed to conduct cost estimates for the autonomous vessel, which are presented in the following sections.

The reference vessel's main and auxiliary engines use marine gas oil (MGO) as fuel, which is the main reason why this particular vessel was chosen. As opposed to small feeders, ships used in intercontinental trades with larger capacity more commonly

Table 2.3: Specifications of the conventional reference vessel (Damen, 2017).

Length Overall [m]	140.6
Length B.p.p. [m]	130.0
Beam [m]	21.8
Depth [m]	9.8
Draft [m]	7.33
Deadweight [t]	9,300.0
Main Engine [kW]	6,000.0
Auxiliary [kW]	850.0
Design speed [kn]	17.0

use heavy fuel oil (HFO) as their main fuel source. This is partially due to the lower price of HFO compared to distillates such as MGO. For instance, in April 2016 the price of MGO was nearly double the price of HFO (Marquard & Bahls, 2015). In addition, merchant ships are governed by the International Maritime Organization's (IMO) marine pollution (MARPOL) conventions, such as Sulphur Emission Control Areas (SECAs). These areas include the Baltic Sea, the North Sea, the English Channel, the United States Caribbean Sea and the coasts along the United States and Canada. As of January 2015 the maximum allowed sulphur content in marine fuels in SECAs is 0.1% by mass (Saxton, 2016). MGO has a sulphur content of approximately 0.1% by mass, and is thus allowed to use in SECAs. HFO on the other hand is not allowed, with a sulphur content of about 1% by mass. Industry insiders claim that MGO will be used more frequently in the coming years, mainly due to the environmental benefits of MGO, and the falling production of residual fuel, or HFO (Marquard & Bahls, 2015).

2.4.1 Operating Costs

Crew Costs

As discussed in Chapter 2.1.1 it's no longer necessary with a onboard crew on the autonomous feeder. Although temporary onboard maintenance work may occur in reality, it is assumed for simplicity that onboard crew costs are negligible. According to operating cost data provided by Drewry (2010) the manning costs constitute 48% of operating costs. These figures include crew wages and overtime payments. The average crew cost share of daily operating costs for the TEU classes listed in Table 2.1 is 48%. This results in a reduction of operating costs by 52%, or USD 97.7 per hour on average.

Shore Control Centre

The Shore Control Centre (SCC) is still on a conceptual level, but the MUNIN project has developed an organizational layout of a SCC capable of monitoring up to 90 vessels simultaneously (Porathe et al., 2014). This layout serves as the basis for estimating costs associated with the SCC from a shipowners perspective. As discussed in Section 2.1.1 the autonomous vessel requires around the clock monitoring from the SCC by navigational operators. According to Kretschmann et al. (2015) the estimated annual personnel cost per ship monitored by the SCC is estimated to be USD 116,000, or USD 13.2 per hour.

2.4.2 Voyage Costs

Autonomous ships are probably not able to operate on HFO (Willumsen, 2018). This is due to the required onboard heating and purifying process of HFO before it can be used. Studies show that this process is inappropriate to automate, meaning that autonomous vessels are restricted to use higher grade fuels such as MGO (MUNIN, 2016). Based on this reasoning it is assumed in this analysis that autonomous vessels are restricted to only use MGO or even higher grade fuels, which will lead to increased voyage costs.

Air Resistance

As presented in chapter 2.1.2, autonomous ships will probably lose major parts of the superstructure, if not all of it. It's conceivable that the autonomous vessel may have some space for a small maintenance crew, but the space is not bound to have sight restrictions like on the bridge on a conventional vessel. Losing the superstructure will reduce the vessel's air resistance. In calm weather air resistance constitute about 2% of total resistance, but is probably significantly higher when facing strong head winds (MAN, 2014). The air resistance faced by the ship in the longitudinal direction is basically a function of the ship's speed and cross-sectional area above the waterline:

$$R_w = \frac{1}{2}\,\rho\;C_d\;V_{app}^2\;A_{cs}\;[\mathrm{kN}] \tag{2.1}$$

where R_w is the air resistance force, ρ is the density of air, C_d is the wind resistance coefficient, V_{app} is the sum of the ship's speed and true wind speed, and A_{cs} is the cross-sectional area of the ship above the waterline. For simplicity the weather conditions are assumed to be calm, which corresponds to a wind speed of about 0.2 $\frac{\text{m}}{\text{s}}$. Based on the dimensions of the reference vessel with capacity of 800 TEUs, A_{cs} was estimated to be approximately 413.6 m^2 (Damen, 2017). By losing the superstructure the area reduced by 75.6 m^2, as illustrated in Figure 2.8.

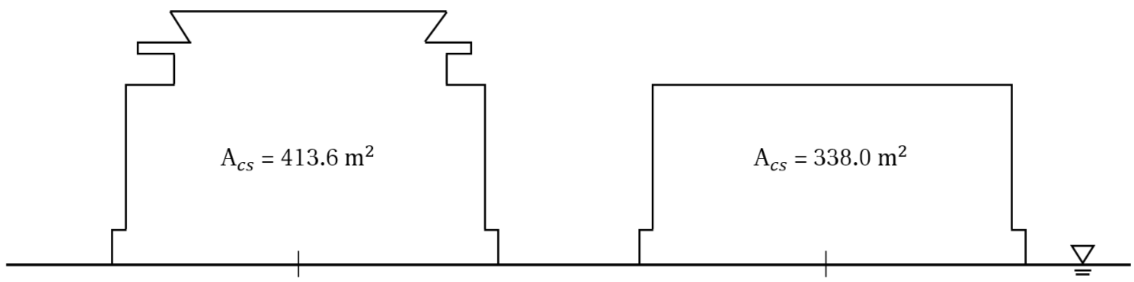

Figure 2.8: Estimated cross-sectional area above the waterline of the conventional and autonomous vessel.

Blendermann (1996) suggests a wind resistance coefficient, C_d, of 0.8 for a typical container vessel. Kretschmann et al. (2015) use a C_d of 0.45 specified for a car carrier with closed fore section, to resemble their autonomous dry bulk carrier. In order to compensate for the container load on deck, this figure is adjusted to 0.55. With an assumed air density of 1.225 kg/m^3, the resulting air resistance forces were calculated according to Equation 2.1. The air resistance force acting on the autonomous vessel decreased by approximately 44%, as summarized in Table 2.4.

Table 2.4: Comparison of wind resistance coefficients and air resistance forces for the conventional and autonomous vessel.

	C_d [−]	R_w [kN]
Conventional	0.80	16.23
Autonomous	0.55	9.12

The reduced air force will reduce the propulsion power demand for the autonomous vessel. The reduction in power can be estimated by using the following formula as presented by Kristensen et al. (2013):

$$P_E = R_w \ V \ [\text{kW}] \tag{2.2}$$

where P_E is the effective propulsion power demand due to air resistance, and V is the vessel's service speed. By using the air resistance values from Table 2.4 in Equation

2.2, the reduction in propulsion power in loaded condition is estimated to be 63.6 kW when the service speed is 17 kn. This corresponds to a total power reduction of about 1% as the reference vessel's main engine has a power output of 6,000 kW. According to Kretschmann et al. (2017) the corresponding reduction in fuel consumption can be assumed to be proportional. Estimating actual fuel consumption is a very complex task as it depend on a range of uncertain and varying factors such as loading condition, draft, speed, waves and currents (Løfstedt et al., 2010).

Light Ship Weight

New autonomous ship designs without superstructure and deckhouse will have reduced light ship weight. Naturally, the reduced light ship weight will influence fuel consumption. Detailed weight information regarding individual sections of merchant ships are rarely published, so in order to calculate the steel weight of the deckhouse and superstructure, appropriate approximation formulas are necessary. Bertram et al. (1998) propose the Müller-Köster method to calculate the steel weight of the deckhouse. As illustrated in Figure 2.9 the deckhouse is split into four separate layers. The weight of each layer is calculated using the volumetric deckhouse weight and deck area relationship proposed by the Müller-Köster method (Bertram et al., 1998). The total steel weight of the deckhouse was estimated to be 252.5 tons.

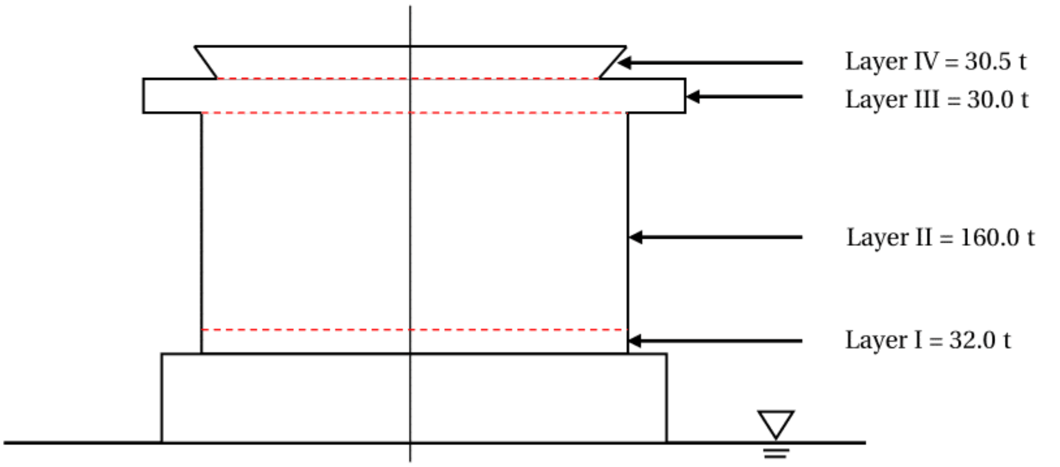

Figure 2.9: Breaking down the deckhouse in individual layers using the Müller-Köster method.

Further, the weight of accommodation and outfitting area, including sanitary equipment, kitchens, walls and inventory, was estimated to be 182 tons. The estimate is based on a recommended density for the accommodation area of 70 m^3 (Bertram et al., 1998) and a approximated deckhouse volume of 2600 m^3. Finally, Bertram et al. (1998) recommend adding 20% of the superstructure and deckhouse steel weight to account for miscellaneous systems below the main deck.

In total the light ship weight reduction of the autonomous ship was estimated to be 485 tons. As the light ship weight of the reference vessel is approximately 7,700 tons, the total reduction amounts to 6.3%. ABS (2014) suggests that fuel consumption will decrease by approximately 0.32% if the steel weight of a container feeder vessel is reduced by 1% while the block coefficient is adjusted such that the deadweight is kept constant. According to this estimation method the reduction in total fuel consumption amounts to approximately 2%.

Electric Power Consumption

As discussed above, the accommodation areas on autonomous vessels are greatly reduced, if not removed completely. In addition to reduced light ship weight, this will result in a reduction of required electrical power from auxiliary systems associated with accommodation and the vessel's hotel system. In order to estimate the impact on total fuel consumption, it's necessary to explore the electrical power balance for a conventional vessel. Kretschmann et al. (2017) use an electrical power balance for a container vessel as a basis for their analysis, as listed in Table 2.5. It's assumed that electrical power for air conditioning, galley and laundry is no longer required on the unmanned vessel, while required power for lighting is reduced by 50%. As presented in the last column, this sums to a total connected load reduction of 40%. This might seem like a substantial reduction, but the impact on total fuel consumption is only approximately 3%. To draw this conclusion it is estimated that auxiliary systems constitute about 9% of total fuel consumption. This estimate is based on fuel consumption data from a range of container feeder vessels with different cargo capacity.

Table 2.5: Electrical power balance from auxiliary systems for a container vessel (Kretschmann et al., 2017 and Mau, 1984).

Power source	Nominal power			
	Total [kW]	At sea [kW]	Unmanned [kW]	Reduction [%]
Propulsion	1,168.0	403.9	403.9	-
Ship operation	142.8	76.6	76.6	-
Air conditioning	374.3	309.3	0	100
Galley and laundry	178.6	138.4	0	100
Deck machinery	609.5	137.5	137.5	-
Cargo ventilation	49.6	43.5	43.5	-
Lighting	91.0	81.0	40.5	50
Other	42.2	37.0	37.0	-
Total load	2,656.0	1,227.2	739.0	40

2.4.3 Capital Costs

Production Costs

As discussed in Chapter 2.2, capital costs represent all expenses related to ship acquisition. Capital costs essentially consist of the new building price of the ship, and the production and assembly costs at the shipyard. Detailed newbuilding prices for autonomous vessels are very difficult to determine as the concept is still in an early stage. However, it is possible to obtain rough production cost estimates using empirical methods. These approximation methods are based on experience from construction of similar ship types, and the cost estimates are calculated by using unit costs. Amdahl et al. (2014) state that it's appropriate to divide production costs into three main categories:

- Hull costs
- Machinery costs
- Outfitting costs

The unit costs presented in Table 2.6 cover the above production cost categories, expressed in terms USD per ton steel, USD per brake horsepower (BHP) and USD per crew member onboard, where assembly and installation costs are included in the unit costs. These costs are considered to be rough estimates, and in an actual contractual agreement between the shipyard and shipowner, the building costs must be more detailed and accurate (Amdahl et al., 2014). However, due to conceptual nature of autonomous ships, these estimates seem like a sufficient starting point.

Table 2.6: Estimated ship production unit costs (Amdahl et al., 2014).

Cost per ton steel	1,900 - 4,300 [$/t]
Cost of main machinery (installed)	306 - 430 [$/BHP]
Cost of auxiliary machinery and machinery equipment	100 - 135 [$/BHP]
Fittings and equipment for crew	62,000 [$/crew member]

To estimate the total capital cost difference between the conventional and autonomous vessel, the calculated change in required power and the reduced light ship weight (chapter 2.4.2) are used. As shown in the last row in Table 2.7 the estimated total building cost for the autonomous vessel is approximately USD 713,045 lower than for the conventional manned vessel. It's assumed that building costs related to crew accommodation and equipment are zero for the autonomous vessel. This may not be the case in reality, as the autonomous vessel might need some space for a short stay maintenance crew. However, in the second last row costs related to autonomous systems are added to the autonomous cost structure. These costs include the autonomous systems presented in chapter 2.1.1, such as ASM, RMSS and AEMCS, and additional costs for other redundant onboard systems (Kretschmann et al., 2015). It is assumed that this figure constitutes 7% of total building costs, or USD 551,025.

Table 2.7: Estimation of capital costs for the conventional and autonomous vessel.

		Conventional		Autonomous	
Cost category	Unit cost	Factor	Cost [USD]	Factor	Cost [USD]
Steel weight	3,000	1,659	4,997,000	1,611	4,833,000
Main machinery	350	8,157	2,854,950	8,054	2,818,900
Auxiliary systems	110	3,331	366,410	1,999	219,890
Fittings and equipment for crew	62,500	15	937,500	0	0
Autonomous systems	-	0	0	-	551,025
Total			9,135,860		8,422,815

As highlighted in MUNIN's quantitative assessment conducted by Kretschmann et al. (2015) a detailed estimation of capital costs should be based on a full ship design, which is not available at this time. Furthermore in the assessment it is assumed that autonomous ship technology constitute about 10%, or USD 1,700,000, of total building costs. In their case with a panamax bulker, this leads to higher building costs for the autonomous vessel than for the conventional vessel. However, it's stated that this might be a rather conservative estimate, and that the opposite case might actually be the correct outcome. As presented in chapter 2.1.2, Rolls-Royce claims that their new autonomous concept design, Electric Blue, is approximately USD 3,000,000 cheaper to build than a conventional vessel (Wilson, 2017). According to Wilson (2017) this figure is even verified by an independent third party. Having this in mind, a considered reduction of USD 713,045 in this analysis seems like reasonable estimate.

Chapter 3

Methodology for Fleet Evaluation in Different Contexts and Needs

In this chapter the structured approach for the analysis is presented. The goal of this chapter is to describe how conventional and autonomous container vessels are compared and evaluated in different contexts and needs. In addition, relevant theory is explained to give the reader insight in the methodologies used. The structured approach in this analysis can be divided into two distinct parts. The first part of the analysis concerns the Maritime Transportation Planning Problem (MTPP), namely the optimal assignment of ports and shipments to vessels in a particular fleet. Later, in the case study in Chapter 6.1, the approach is applied to a specific trade with a given number of ports with individual container cargo demands. The desired output of this part of the analysis is operational and voyage costs for several fleet configurations, both conventional and autonomous. The cost estimates presented in chapter 2.4 serve as the basis for the desired cost outputs, and the underlying cost regression models are able to change when the context change.

The second part of the analysis focuses on how to quantify changeability, and aims to answer the following questions:

- What happens to costs when the context change?
- What fleet configuration minimizes cost and performs best in different contexts?

3.1 Maritime Transportation Planning Problem

The first part of the analysis deals with solving a Maritime Transportation Planning Problem (MTTP) where the desired output is the optimal visiting sequence and assignment of shipments to the vessels in the fleet. This results in a total distance the vessels have to travel to serve all ports and their respective demands. The total distance is used in the regression model to calculate capital costs, voyage costs and operating costs. Before the mathematical model is presented in Chapter 5, relevant theory is presented in the following sections.

Merchant ships are mainly operated in three different shipping modes: *liner*, *industrial* or *tramp* mode. In liner mode the vessels follow a fixed route according to a schedule and a fixed tariff, just like a bus service (Fagerholt et al., 2012). Its itinerary is published which attracts demand. The majority of the shipped cargo in liner mode is containerized general cargo. The vessels follow pre-determined timetables published several months ahead with a frequency of sailing that vary with season. The routes may stay fixed for several years. Tramp ships do not follow fixed routes, but follow the available cargo. Tramp ships typically engage in contracts of affreightment where specified cargoes are to be transported to its destination within an agreed upon time period. Tramp operation is therefore compareable to a taxi-service. In industrial mode the operator owns the cargo and controls the ships, either by chartering or owning the ships.

As an owner of a merchant cargo ship there are a variety of different scenarios or problems that you are likely to encounter. In the following the most relevant types of problems are defined. *Routing* is defined as the assignment of ports, i.e. which port to visit, to a vessel. Often in conjunction with routing, *scheduling* is the process of assigning times, commonly called time-windows, to the port visits on a vessel's route. This is a common issue in shipping as perishable cargo need to be transported within a certain time. A *fleet size and mix* problem involves to decide the optimal number of vessels in a fleet (size), and the optimal mix of vessels in a fleet with equal or different specifications (e.g. cargo capacity, length, draft). A fleet is said to be *homogeneous* if all of the vessels have the same specifications, and *inhomogeneous* if the vessels have differing specifications.

MTTPs are traditionally divided into three main planning levels; *strategic*, *tactical* and *operational* (Christiansen et al., 2007). The different planning levels are categorized according to the planning horizon and by the desirable problem output. Even though the planning problems are divided into separate categories, it's important to note that the planning levels can be highly interrelated. Strategic problems have a planning horizon of 1 to 20 years and typically involve problems such as:

- Fleet size and mix decisions
- Contract evaluation
- Ship design
- Network design

Tactical problems have a planning horizon of 1 week to 1 year, and include problems such as:

- Fleet deployment
  - Assignment of vessels to routes
- Ship routing and scheduling
- Container stowage planning
- Distribution of empty containers

Operational decisions have a time frame of 1 day to 1 week, and typically involve:

- Environmental routing
  - Weather routing
  - Ocean currents
- Speed selection
- Cargo loading
- Single order bookings in liner shipping

The maritime transportation problem in this thesis is on a tactical and strategic level. The tactical part of the problem is a routing problem, i.e. the assignment of a sequence of ports to a vessel (Christiansen et al., 2007). In addition, the problem involves assigning vessels to specific shipments, which classifies as scheduling. The strategic part of the problem is a fleet size and mix problem, i.e. to determine the optimal capacity for each vessel and the optimal number of vessels in the fleet.

3.1.1 Mathematical Optimization

Regardless of the planning level, the real world problem has to modeled in mathematical terms. More specifically, the problem has to be modeled as an mathematical *optimization* model. An optimization process consists of four distinct phases; *identify*, *formulate*, *solve* and *evaluate*. The real world problem is often complex, so the first step in an optimization process is to *identify* the most relevant factors to simplify the problem. In this case this would be to identify the elements in the MTTP that affect autonomous and conventional container vessels. As visualized in Figure 3.1, the outcome is a simplified version of the real world problem.

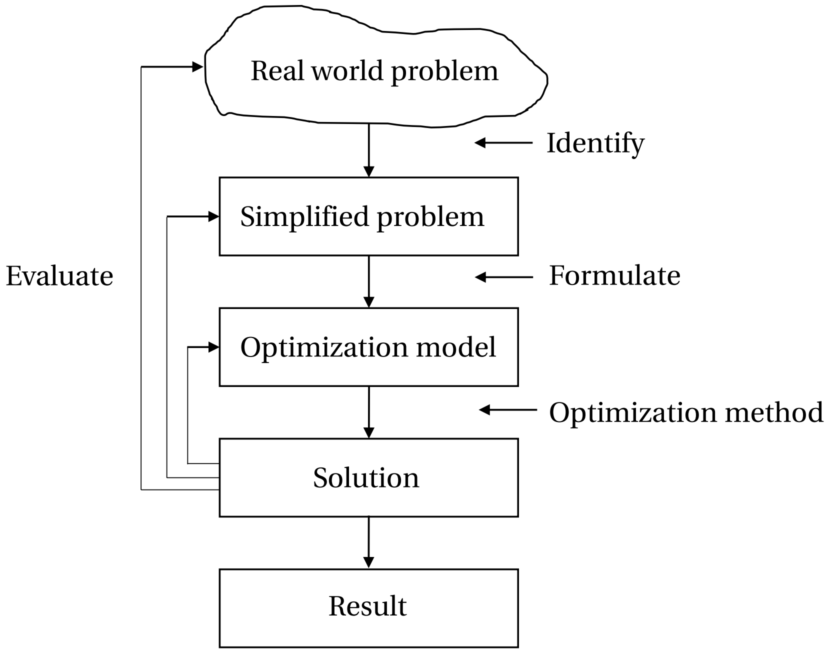

Figure 3.1: The optimization process (reproduced from Lundgren et al., 2010).

The next step is to *formulate* the simplified problem as a mathematical optimization model. Let's say there are n decisions to be made in the problem, they are defined as *decision variables*; $x_1, x_2, \ldots, x_n$. The decision variables are quantifiable, and the goal of the problem is to determine their respective values (Hillier et al., 2001). The *objective function* is an expression of the desired measure of performance, e.g. cost; $C = 2x_1 + 5x_2 + \cdots + 8x_n$. The objective function is then maximized or minimized subject to one or several *constraints*; e.g. $x_1 + 2x_2 \leq 20$.

In this analysis the objective is to minimize costs, which is reflected in the distance travelled by all the vessels in the fleet. The coefficients and the right-hand side in the objective function and constraints are defined as the *parameters* of the optimization model. This is the most basic form of an optimization model, but it can be far more complex depending on to what extent the real world problem is simplified. The mathematical optimization model in this analysis is presented and explained in in detail Chapter 5.

A mathematical optimization model can be solved either analytically or numerically by commercial computer solvers. An analytical solution provides the *true optimal* solution since all possible solutions are compared. However, when the complexity of the problem increases it can be extremely difficult to find the true optimal solution. To illustrate how complex the solution can get, the Travelling Salesman Problem (TSP) is used as an example. The TSP is a well known problem in operations research, where the goal is to find the shortest path between a set of nodes such that all nodes are visited exactly once, and the start and end node are the same. This is called an *Hamiltonian* path, as illustrated in Figure 3.2. In a set of $n = 7$ nodes there are $(n-1)! = (7-1)! = 720$ possible Hamiltonian paths to consider. If the computer solver can compare 1 billion unique Hamiltonian paths per second the TSP is solved in approximately 0.72 microseconds. However, if the number of nodes increases to $n = 20$ it would take about 3.9 years for the computer to compare all possible solutions.

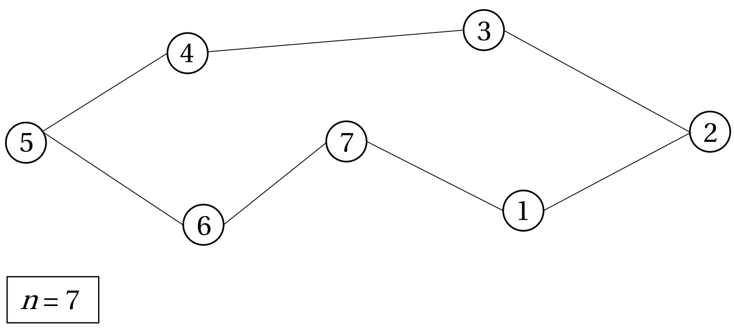

Figure 3.2: Hamiltonian path.

To save computational time the mathematical model can be solved numerically by using rule based solutions algorithms called *heuristics* (Norstad, 2017). When using a heuristic there is no guarantee that the solution is optimal, but hopefully it provides a sufficient solution that is close enough to the optimal. Heuristics are case-sensitive, meaning that the type of algorithm used must fit the actual problem at hand. The routing problem in this analysis is called a vehicle routing problem (VRP), which is explained in the following section. The problem is partially solved by a method called route generation, which involves to generate all feasible routes for each individual vessel. The set of feasible routes are then used as input to solve an Integer Programming (IP) model, i.e. determine the routes with lowest cost (distance). The model is said to be an IP model if the solutions (variables) are restricted to be integers, which is the case for the model in this analysis. An IP model is easier to solve because of its simpler structure than the direct mathematical formulation. Furthermore, it's easier to include practical constraints such as time-windows, capacity, etc. However, the route generation method is a two step approach since all feasible routes must be generated in advance to ensure optimal solutions. As the number of feasible routes grows with problem size, the computational time will increase accordingly. The route generation procedure and mathematical model in this analysis is explained in more detail in Chapter 5.

3.1.2 Vehicle Routing Problem

The vehicle routing problem (VRP) is a generalization of the travelling salesman problem (TSP), or the multiple-TSP, meaning multiple ships in this case. More specifically, a VRP is the assignment of routes from a depot, or several depots, to a set of nodes, or ports. The VRP is applicable to a range of different scenarios different from maritime transportation. The overall goal of the VRP is to determine the optimal visiting sequence, or route, for each vessel in the fleet, subject to vessel capacity constraints. As illustrated in Figure 3.3, each port has an individual demand, denoted by a number next to the respective node. The problem can be formulated as a *pickup* or *delivery* problem. The former is the situation when the cargo is shipped from the nodes to the depot, and the latter is the opposite situation. A delivery problem is equivalent to a Hub & Spoke network structure where the hub is the depot, and the spokes represent the ports. The problem can also be formulated as a pickup and delivery problem where the cargo flows between spokes and the hub. In the example shown in Figure 3.3 there are three vessels in the fleet, as the vessels are assigned to one route each.

The VRP model can be extended with several constraints to achieve a more realistic model at the expense of higher complexity. Common VRP extensions are listed in Table 3.1. The VRP model in this analysis is time constrained, i.e. all of the container cargo must be shipped within a certain time period. As opposed to the example illustrated in Figure 3.3, the VRP model in this analysis is classified as a Multi-Trip VRP,

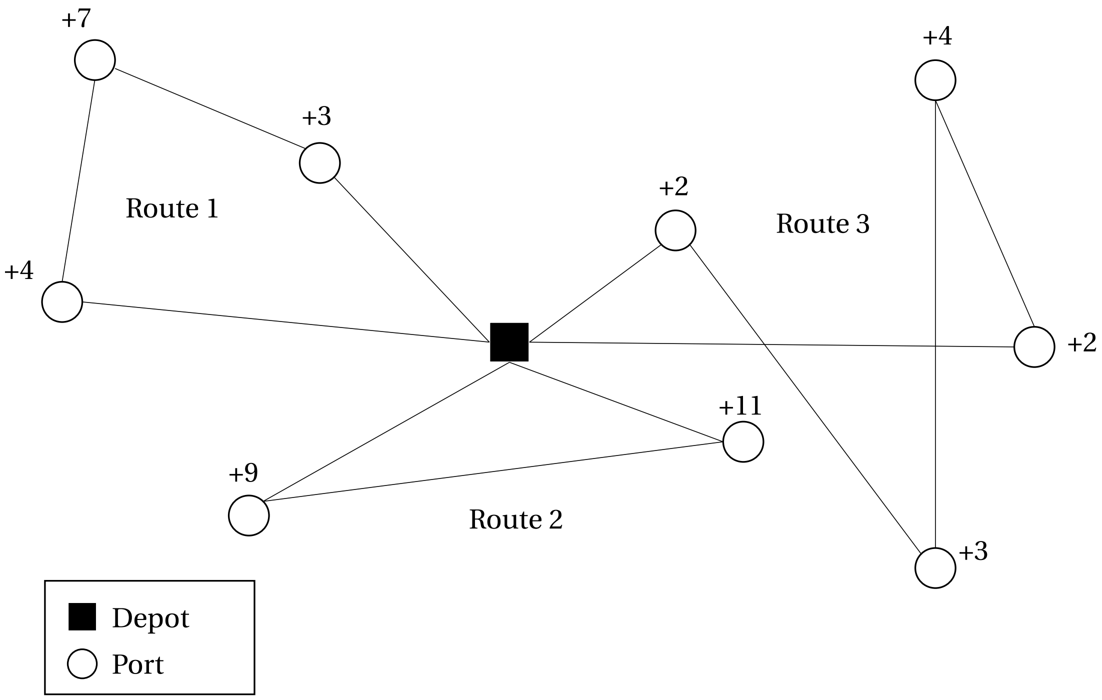

Figure 3.3: Example of a typical VRP.

meaning that each vessel in the fleet is allowed to make multiple trips. This may be the case if a particular ship in the fleet has lower route cost, or if the remaining ships are constrained by capacity for a particular route. The model is classified as a delivery problem, meaning that the cargo is moved from the hub, or depot, to the respective ports. The network consist of a single depot, and there are no compatibility or precendence constraints considered.

Table 3.1: Typical VRP extensions.

VRP extension	**Characteristics**
Max. duration of a route	Work day limitations
Compatibility	Port compatibility, i.e. draft, length and breadth constraints
Time windows	A time interval when service is required
Precedence constraints	Some customers must be visited before others
Pickup and delivery	Each customer has one pickup node and one delivery node
Multi-Depot	Several depots
Multi-Trip	More than one tour per vehicle allowed

3.2 Contextual and Temporal Design Approach

The second part of the analysis focuses on the dynamic aspect of system design. The goal of this part of the analysis is to measure the performance of autonomous and conventional container vessels when the static context change. In traditional system design the approach is to optimize a system according to a set of objectives in a static context. The output of this design approach, namely the optimal system, is largely dependent on that the assumptions made in the analysis stay static in operation. If the parameters or variables that define the system change over time, there is no guarantee that the system will remain optimal (Ross et al., 2008b). As the real world is dynamic, the traditional design approach may often fall short. Studies conducted by Carlson et al. (2000) show that "over-optimized" systems are fragile in a dynamic context. This may be a plausible claim as a system is traditionally designed for optimality in the context it is designed for, and may not perform as well in other contexts.

To *design for value robustness* as highlighted by Ross (2006) is a system design approach where the main focus is to search for system solutions that will continue to perform well in dynamic operational contexts. Value in this setting is a measure of success defined by the system stakeholders, and the best design is the design that creates most value over time. Ross et al. (2008) suggest that in order to design value robust solutions, it is necessary to move beyond traditional engineering and optimization methods, and shift focus towards temporal factors affecting the system. In the following sections, two such paradigms are presented; tradespace exploration and epoch-Era analysis (EEA). Lastly, a step-by-step design method that utilizes tradespace exploration and EEA, called the responsive systems comparison (RSC) method, is presented.

3.2.1 Tradespace Exploration

Tradespace exploration is a useful visual comparison method for evaluating possible design solutions, formulated by Ross et al. (2005). A typical tradespace is a plot of a substantial amount of different system designs. The system designs are commonly assessed in terms of cost and utility. In this case, utility is a direct reflection of perceived value, and is determined by the decision makers (Ross et al., 2006). Figure 3.4 illustrates a tradespace representation for the design space, i.e. all design solutions, of an anchor-handling tug supply (AHTS) vessel. The scatter plot represent all possible design solutions with corresponding cost on the horizontal axis, and utility on the vertical axis.

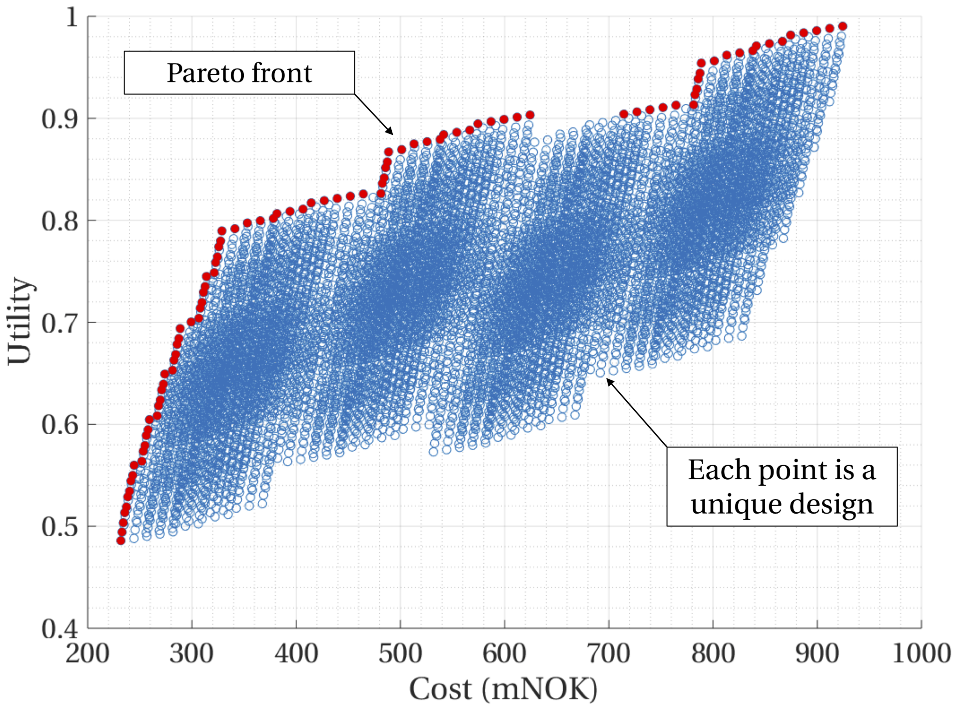

Figure 3.4: Tradespace representation of several possible design solutions of an AHTS vessel.

Each point in the tradespace correspond to one unique design solution which is defined by designer-specified metrics such as length, capacity, fuel consumption, etc. As opposed to the designer, the decision maker will eventually decide between design solutions by using utility as the key metric. The points marked in red are said to be on the *Pareto Front*, i.e. the best designs in this particular context. A set of pareto designs has the highest utility for a particular cost. Choosing between these designs involves making trade-offs between cost and utility. The designs that fall outside the Pareto front are called *dominated* designs.

One of the key benefits with the tradespace approach is that any change that affect cost or utility, can be quickly assessed in a new tradespace. Figure 3.5 shows an example of a tradespace impact when the system utility functions have changed. The decision-maker can now easily inspect the designs that are sensitive or insensitive to the given change in preference. The marked points in Figure 3.5 represent designs that shifted in opposite direction of the change in preference (utility), nor with the same magnitude. Changes in design parameters can also be captured in the tradespace, as any parameter change may affect cost and utility.

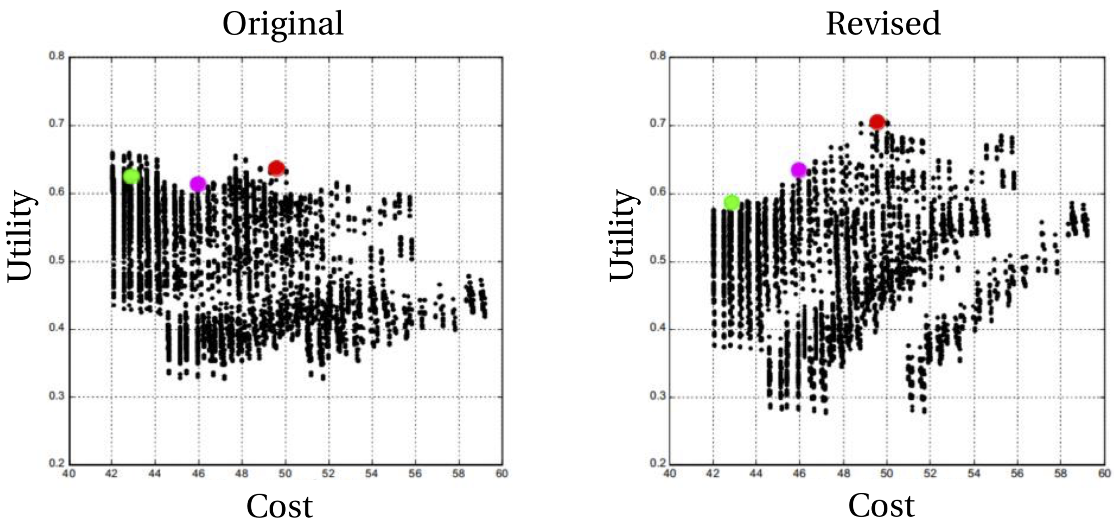

Figure 3.5: Tradespace impact of change in utility (reproduced from Ross et al., 2005).

3.2.2 Epoch-Era Analysis

In EEA the system's lifespan is divided into a sequence of different time periods, or *epochs*. The contextual factors that form an epoch are static within each individual epoch. Several epochs in a time-ordered sequence is defined as an Era, which represents a dynamic interval of time (Gaspar et al., 2012). Ross et al. (2008) use the analogy of a movie composed of a series of static frames in sequence to describe the connection between static and dynamic time intervals. As illustrated in Figure 3.6 an epoch represents a static frame, while several epochs in sequence, or an era, constitute a dynamic perspective.

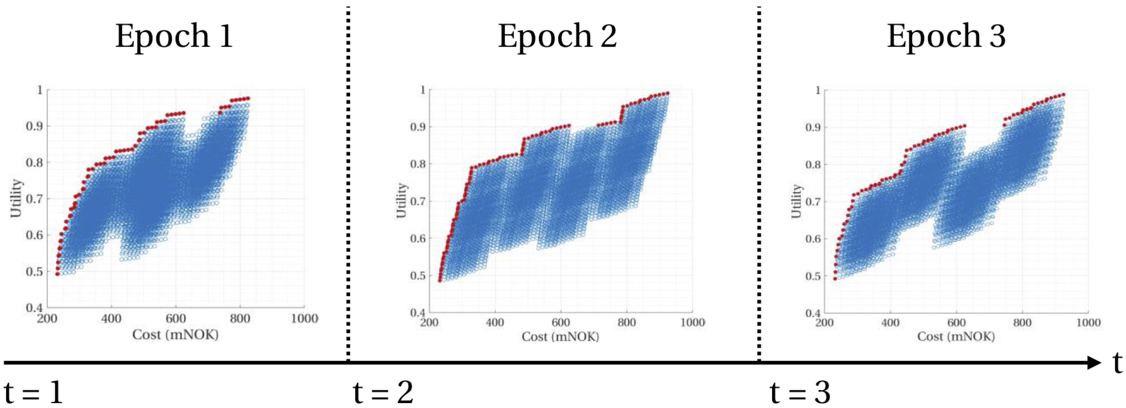

Figure 3.6: Epoch-Era Analysis (EEA) visualized as static time intervals in sequence.

The entire lifetime of system is referred the as the *System Era*. A change in contextual factors triggers the start of a new epoch, e.g. a change in oil price, a new market condition or new environmental regulations. When the system experience change, it may need to transform in order to meet expectations and sustain value, which is visualized in Figure 3.6 where the tradespace shifts in each epoch.

3.2.3 Responsive Systems Comparison Method

The Responsive Systems Comparison (RSC) method is a system design method where the purpose is to "guide a designer or system analyst (RSC practitioner) through a step-by-step process of designing and evaluating dynamically relevant system concepts" (Ross et al., 2009). The RSC method is appropriate when dealing with a substantial amount of possible system designs to consider, called the *design space*. Ultimately, the RSC practitioner will have insight in the trade-off between different design solutions, and the performance of the design space in different contexts. The original RSC method consisted of seven steps as presented first by Ross et al. (2009) and Ross et al. (2008). However, in more recent times the method has evolved to a nine step process, as illustrated in Figure 3.7.

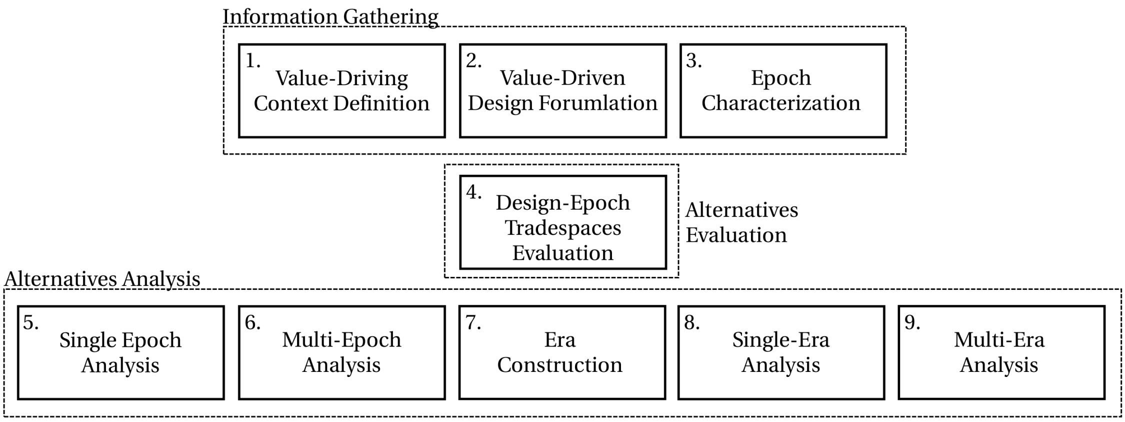

Figure 3.7: Steps of the Responsive Systems Comparison (RSC) method (adapted from Schaffner et al., 2014).

An adapted version of the RSC method is used in the analysis in this thesis. The method is appropriate because it deals with changes in needs, context and system (Ross et al., 2009). The steps in the RSC as illustrated in Figure 3.7 are described in the following paragraphs:

Step 1: Value-Driving Context Definition

The idea behind the first step is to identify the problem statement, and create a value proposition. The most important exogenous factors that might affect the problem and the solution are identified. As pointed out by Pettersen et al. (2017) the value proposition should act as the "link between the scope of the system design process and the business strategy of the stakeholders".

Step 2: Value-Driven Design Formulation

In the next step the specific stakeholder's needs are extracted from the value proposition, called attributes. These attributes are metrics that reflect system performance, and are quantified in terms of *utility* functions. The utility functions describe the stakeholder's preference for each of the extracted attribute (Schaffner et al., 2014).

Step 3: Epoch Characterization

In the third step, contextual uncertainties are parameterized and defined according to the stakeholder's expectations (Schaffner et al., 2014). These discrete variables are called *epoch variables*. Each combination of the epoch variables constitutes one particular contextual setting, or epoch. If the epoch variables are chosen to be oil price and environmental regulations, a potential epoch would be a period with e.g. high oil price and medium environmental regulations. Within each epoch the epoch variables are not allowed to alter, and an epoch is defined as a static short-run scenario (Pettersen et al., 2017).

Step 4: Design-Epoch Tradespace Evaluation

All possible design solutions are plotted in terms of cost and utility, as previously shown in Figure 3.4. This makes it possible for the RSC practitioner to easily observe the trade-offs between utility and cost, and the Pareto frontier designs.

Step 5-9: Alternatives Analysis

The last steps of the process deal with construction of eras, and analysis of design solutions across several epochs and eras. In step 5, Single-Epoch Analysis, individual epochs are analyzed in a similar manner as in step 4 to identify the designs with best performance.

Furthermore, in Multi-Epoch Analysis, multiple epochs are analyzed to identify design solutions with high performance when the context change. For instance, which designs have high utility and low cost in epochs with low, medium and high oil price?

In order to analyze the system's lifetime performance, epochs are structured in sequence to construct eras in step 7. As opposed to epochs, eras represent the long-run perspective of the system. As for the Single Epoch-Analysis, eras are analyzed individually before multiple eras are constructed to identify patterns across eras by analyzing dynamic system properties (Pettersen et al., 2017).

Chapter 4

Literature Review

This chapter presents a review of relevant literature related to the approach and scope of this thesis. In the first part of this chapter research concerning relevant routing and scheduling problems are reviewed and discussed. Next, the focus shifts towards literature that utilize the RSC method, and consequently EEA, in maritime system design. The last part is devoted to research concerning cost assessments of autonomous ships and autonomous infrastructure.

4.1 Routing and Scheduling

In general, transportation planning is a commonly discussed subject, but the majority involves research about transportation by air and road (Christiansen et al., 2007). Other transportation modes, such as seaborne transportation, has been devoted far less attention. As pointed out by Christiansen et al. (2007) the reasons for this might be low visibility of ships (i.e. most people see trucks, planes etc.), the diversity of maritime planning problems, and higher uncertainty in the shipping market compared to other transportation modes. Historically, the shipping industry has a long tradition and is fairly fragmented, due to strong ties in family owned companies. However, there has been a large increase in research concerning maritime transportation the last decade (Fagerholt et al., 2012).

Maritime routing and scheduling is a widely reviewed subject in recent literature. Ronen (1983), Christiansen et al. (2004) and Fagerholt et al. (2012) are some notable publications that review the current research within the field of routing and scheduling. The latter presents a thorough review of routing and scheduling problems in the new millennium, where four basic models are presented. Fagerholt et al. (2012) state that the research volume regarding routing and scheduling almost doubles every decade. This recent trend is probably connected to the current market trends, where advanced data analytics and optimization to support decision making have gained attention. For instance, the increase in container trade has caused a similar increase in liner network design research.

The review survey conducted by Fagerholt et al. (2012) classify the models according to mode of transportation, namely liner, industrial and tramp shipping. Further, liner shipping models with sets of routes without transshipment are reviewed. In this context "models without transshipment" mean that transshipment costs are neglected, which is the case for the model in this thesis.

Network Design Models with Sets of Routes without Transshipment
Fagerholt (2004) considers a network design problem where the goal is to design optimal weekly routes for a given heterogeneous fleet of container vessels. The network is a feeder system consisting of several spoke ports and one hub located in Norway. The problem is modeled as a Multi-Trip VRP, and solved by generating a set of feasible routes for each vessels in the fleet. By feeding the feasible routes to an integer programming (IP) model, the optimal routes for each individual vessel that minimize total operational cost for the entire fleet are determined. The problem is modeled with time constraints, i.e. the vessels may sail several routes unless the time constraint of one week is violated. The model was applied on a range of different test problems, with the largest network size of 40 spoke ports and 20 vessels. Fagerholt (2004) concludes that the proposed method works and that the problem is solved to optimality.

Sigurd et al. (2005) formulate a network design model for a set of routes located in Norway and Amsterdam. The problem is time constrained to one week, where several visits each week are allowed. As opposed to this study, the fleet consists of high-speed RoRo vessels. In addition, minimal time separations between port calls are considered. The model is solved by pregeneration of feasible weekly routes and column generation.

Chen et al. (2010) formulate a mixed integer non-linear programming model for a container shipping network. The model is solved using a bi-level genetic algorithm. In contrast to this thesis, Chen et al. (2010) consider fluctuating demand and empty container re-positioning. The fluctuating demand is artificially generated, but Chen et al. (2010) conclude that their model provide a more realistic solution compared to models based on average demand.

Lun et al. (2009) investigate the impact of fleet mix on cost performance in container shipping. In the first part of the analysis the optimal ship for a set of routes is determined. Then the impact of carrying capacity and ship size (fleet mix) is investigated using path analysis. Lun et al. (2009) find that the cargo capacity has a stronger impact than the actual ship size. The impact of fleet mix is studied in the same manner in this thesis, where the impact on costs due to cargo capacity for individual vessels is assessed.

As indicated by Fagerholt (2004) there are few studies that address feeder hub and spoke networks. The majority of these studies only deal with determining the optimal fleet size when the fleet is fixed, such as Bendall et al. (2001) and Fagerholt (1999).

4.2 Responsive Systems Comparison Method

Pettersen et al. (2017) demonstrate the use of the Responsive Systems Comparison (RSC) method on the design of an industrial offshore construction vessel. The design problem is referred to as an *ill*-structured decision problem since a value robust system typically has properties such as flexibility, agility, scalability, etc. Pettersen et al. (2017) find that the strength of the RSC method compared to classic design methods is the ability to easily observe the trade-off between the cost and value. Furthermore, it is stated that the design problem becomes more tangible as the assumptions are reduced to a minimum. In contrast to this thesis, Pettersen et al. (2017) apply the RSC method on the design of single vessel rather than optimal fleet configuration and cost comparison. The problem is of different nature as the aim in this thesis is to compare fleet performance in different contexts, opposed to determine the most robust design specifications.

Gaspar et al. (2012) utilize Epoch-Era Analysis (EEA) to handle contextual uncertainty related to future contract and market opportunities in the design of non-transport vessels. Similar to the approach in thesis, Gaspar et al. (2012) use epochs to capture the temporal and contextual design aspects. The performance of the vessel in different epochs is measured by solving a Ship Design and Deployment Problem (SDDP) for a set of possible contracts. The output of the SDDP is the optimal contract path that maximize revenue, similar to the routing problem in this thesis. Gaspar et al. (2012) conclude that SDDP combined with EEA is an appropriate and efficient method in initial ship design.

4.3 Quantitative Assessment of Autonomous Vessels

The cost structure for the autonomous vessel presented in Chapter 2.4 is primarily based on the quantitative assessment conducted by the MUNIN project. The assessment is conducted by Kretschmann et al. (2017) and is published as a scientific paper. However, the majority of the estimations and calculations in the paper refers to an in-depth quantitative deliverable to the MUNIN project (Kretschmann et al., 2015). The research compare estimated operating, voyage costs and capital costs of an autonomous bulker to a conventional bulker. The costs are first estimated on an annual basis, and then calculated for a 25-year period using expected present value (EPV). The autonomous vessel is compared to the reference bulker in three different scenarios:

- **Scenario A**
  The first scenario considers the impact of a reduced crew and extra costs for new port services.
- **Scenario B**
  The same as scenario A but increased fuel efficiency is also considered.
- **Scenario C**
  The final scenario is the same as scenario A and B, but now Marine Diesel Oil (MDO) is used as main fuel instead of Heavy Fuel Oil (HFO).

In scenario A, Kretschmann et al. (2017) find that the EPV of costs for the autonomous bulker is almost as the same as the conventional. Considering increased fuel efficiency (scenario B) results in an EPV of costs of USD 4.3 million lower than the reference vessel. However, in scenario C the EPV of cost of the autonomous vessel is USD 19.1 million higher than the conventional vessel that uses HFO as main fuel. Based on the results from these scenarios it's concluded that autonomous vessels are beneficial in terms of cost, if the fuel efficiency is increased. Kretschmann et al. (2017)

emphasize that the results may be somewhat uncertain as autonomous vessels are still in development.

Chapter 5

Model Description

In this chapter the route generation procedure and mathematical model for the multi-trip VRP are presented. The approach is adapted from Kjetil Fagerholt's work in *Designing optimal routes in a liner shipping problem* (2004). The only differences lie in the route generation procedure and input data.

The pipeline solution algorithm in its entirety is illustrated in the flowchart in Figure 5.1. The input to the model is a set of fleets with individual capacities as indicated by *Fleet Input Data*. The solving procedure is divided into two phases. The first phase deals with route generation for each individual vessel with respect to time and capacity constraints. The output of this phase, namely a set of feasible routes, is fed to an integer programming (IP) model in the second phase. The IP model finds the optimal routes for each vessel that minimize transportation costs. In addition, the IP model assures that all ports are visited at least once. In the following, each step in the solution procedure is explained in detail.

5.1 Phase 1: Route Generation

The first step in the route generation phase is to generate all feasible routes with respect to capacity and the time limitation of one week. This is achieved by calculating all possible route combinations and the corresponding demand and time requirements for each route. Then all feasible routes for each vessel are selected. The route generation procedure was conducted in MATLAB, and the reader is referred to

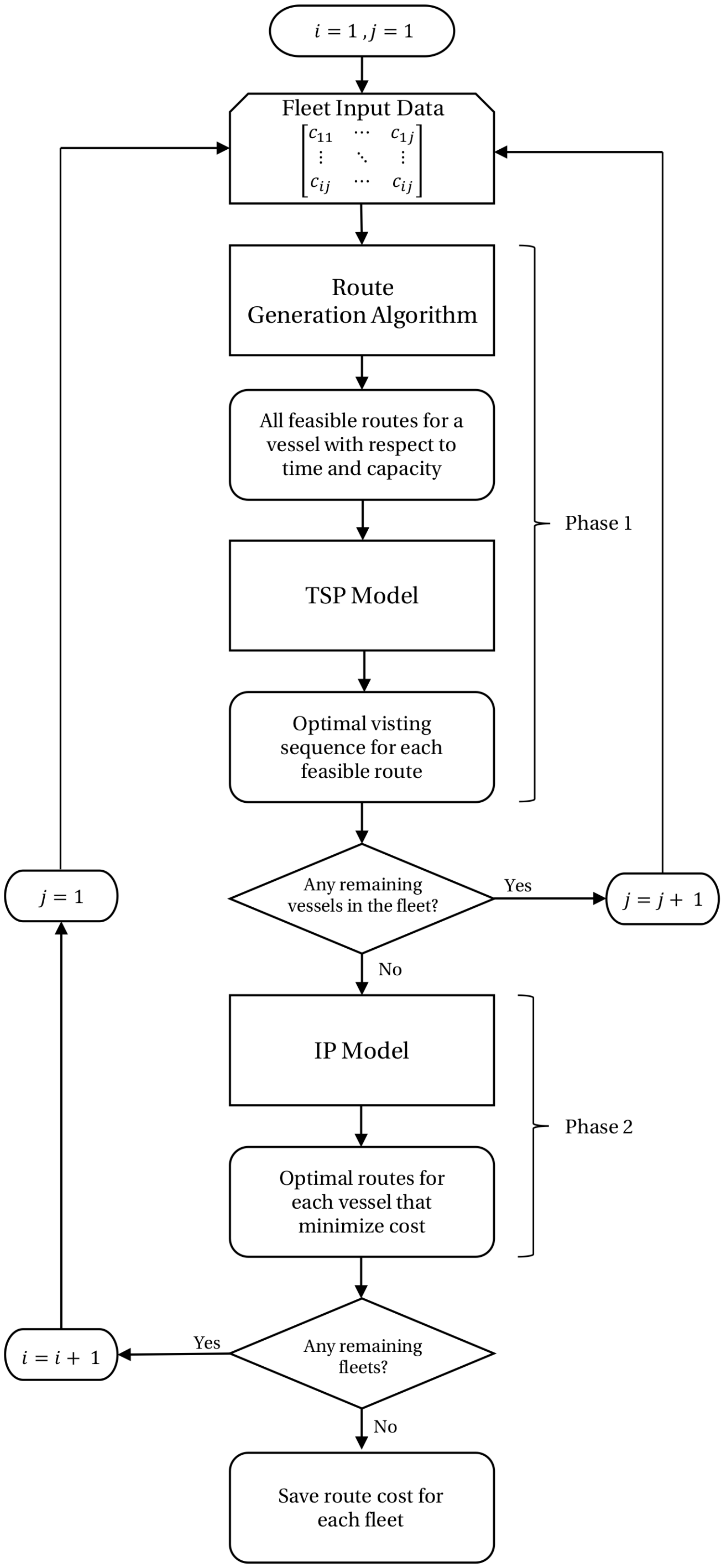

Figure 5.1: Flowchart showing the pipeline solving procedure (adapted from Fagerholt, 2004).

Appendix B.2 for the corresponding MATLAB-script. The following example illustrates the route generation algorithm for a particular vessel with capacity of 500 TEU:

Assume a network consisting of three spoke ports and one hub port as depicted in Figure 5.2. The spoke ports have individual demand, and an associated sailing time (days) represented by the numbers next to the arcs connecting the nodes. The maximum allowed sailing time is seven days.

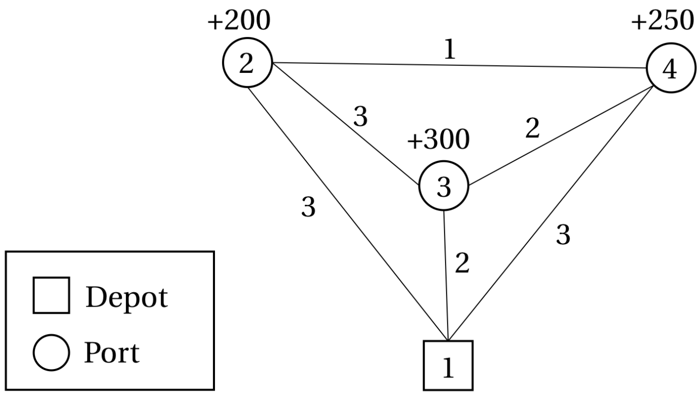

Figure 5.2: Example network (adapted from Fagerholt, 2004).

Table 5.1 shows all feasible routes for the 500 TEU vessel. The route 1 - 3 - 2 - 1 is infeasible because the total sailing time exceeds seven days. If the vessel sails route 1 - 4 - 3 - 1 the required demand is greater than the cargo capacity of 500 TEU, so this route is also considered infeasible.

Table 5.1: Feasible routes for the example problem (adapted from Fagerholt, 2004).

Route ID	**Nodes**	**Sailing time**
1	1 - 4 - 2 - 1	7
2	1 - 3 - 1	4
3	1 - 4 - 1	6
4	1 - 2 - 1	6

The next step in phase 1 is to determine the optimal visiting for the feasible routes. This is achieved by solving a TSP, as presented in Chapter 3.1.1. In this thesis all feasible routes with more than two spoke ports are fed into the TSP model and solved one by one. As opposed to the procedure presented by Fagerholt (2004) where a feasible route is extended by a port *i*, and then a TSP is solved with the initial nodes and node *i*. The results of the procedures are essentially the same, but the procedure formulated by Fagerholt (2004) may be more efficient in terms of computational time.

The following mathematical model is used to solve the TSP. The model is implemented and solved in FICO Xpress IVE, and the corresponding script written in Mosel code can be found in Appendix C.1.

5.1.1 TSP Model

In order for the following model to work the distance matrix is restricted to be symmetric. The edges between all nodes are enumerated and indexed by e, and a binary decision variable U_e is introduced:

$$U_e = \begin{cases} 1, & \text{if edge } e \text{ is used} \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

The following indices, sets and parameters are used in the model:

Indices

e Edges

i Nodes

Sets

E Set of edges

N Set of nodes

Parameters

c_e Cost of sailing edge e

The aim of the model is to minimize sailing cost, so the objective function of the TSP model becomes:

$$\min \sum_{e \in E} c_e U_e \tag{5.2}$$

Let $\delta(i) = \{e \in E \mid e \text{ is connected to } i\}$, i.e. the set of undirected edges connected to node i. Then the objective function, Equation (5.2), is subject to the following constraints:

$$\sum_{e\in\delta(i)} U_e = 2 \quad \text{for all } i \in N \tag{5.3}$$

$$U_e \in \{0,1\} \tag{5.4}$$

$$\text{Subtour elimination constrains} \tag{5.5}$$

Constraint (5.3) says that the number of edges connected to each node must be equal to two. This ensures that all nodes are visited exactly once. Constraint (5.4) specify that the decision variable, U_e, is binary. The subtour elimination constrains (5.5) ensure that the solution is a true Hamiltonian tour (see Chapter 3.1.1). A subtour is a tour where the start node and end node is correct, but not all nodes are visited. To eliminate subtours the subsets W end $E(W)$ are introduced. W is a subset of N containing at least two elements and at maximum $n-1$ elements, where n is the total number of nodes. $E(W)$ is defined as the set that contains the set of edges connected to the nodes in W; $E(W) = \{e \in E \mid \text{both ends of } e \text{ exists in } W\}$. The subtour elimination constraint becomes:

$$\sum_{e\in E(W)} U_e \leq |W| - 1 \quad \text{for all } W \subset N,\ 2 \leq |W| \leq n-1 \tag{5.6}$$

where $|W|$ is the cardinality of W, i.e. the number of elements in W.

Referring to the flowchart in Figure 5.1, the route generation (phase 1) is repeated for all vessels in the fleet. The results from the TSP serve as the input for the IP model in the second phase. The input data is processed in MATLAB and transmitted to Xpress IVE via the MATLAB-Xpress interface. The reader is referred to the script `FleetGen.m` in Appendix B.1 for processing of input data and the iterative execution of Xpress IVE via MATLAB.

5.2 Phase 2: Integer Programming Model

The objective of the IP model is to determine the optimal routes for each vessel in the fleet that minimize operational cost. In the selection of optimal routes, the IP model also ensures that all ports are visited. The following indices, sets and parameters are used in the IP model:

Indices

k	Ships
r	Routes
i	Nodes

Sets

A^k_{ri}	A constant equal to 1 if route r for ship k services node i, or 0 otherwise
R^k	Set of candidate routes for ship k
C^k_r	Cost for ship k to sail route r
T^k_r	Elapsed time for ship k to sail route r

Parameters

T_{max}	Maximum allowed sailing time

A^k_{ri}, R^k, C^k_r and T^k_r were determined in phase 1, and serve as the input to the model. The binary decision variable x^k_r is introduced:

$$x^k_r = \begin{cases} 1, & \text{if ship } k \text{ use route } r \\ 0, & \text{otherwise} \end{cases} \tag{5.7}$$

The objective function of the IP model becomes:

$$\min \sum_{k \in K} \sum_{r \in R^k} C^k_r x^k_r \tag{5.8}$$

The objective function is then minimized subject to the following constraints:

$$\sum_{k \in K} \sum_{r \in R^k} A^k_{ri} x^k_r \geq 1 \qquad \forall i \in N \tag{5.9}$$

$$\sum_{r \in R^k} T^k_r x^k_r \leq T_{max} \qquad \forall k \in K \tag{5.10}$$

$$\sum_{r \in R^k} x^k_r \geq 1 \qquad \forall k \in K \tag{5.11}$$

$$x^k_r \in \{0,1\} \qquad \forall k \in K, \forall r \in R^k \tag{5.12}$$

The first constraint (5.9) assures that all ports are visited at least once. Constraint (5.10) ensures that the total sailing time does not exceed the time restriction of one week. The third constraint (5.11) says that all vessels in the fleet must be used. The last restriction specifies that the decision variable is binary.

The output of the second phase is the operational cost for a particular fleet. Referring again to the flowchart in Figure 5.1, the first and second phase are now repeated until all of the fleets are executed. The IP programming model was solved in Xpress-IVE, and the corresponding script is located in Appendix C.2. The output for each iteration is stored in MATLAB by the use of the MATLAB-Xpress interface. The entire solving procedure is executed by running `FleetGen.m` in MATLAB (see Appendix B.1).

Chapter 6

Case Study: Regional Baltic Trade

This chapter presents the case study in which autonomous and conventional container fleets are evaluated and compared. The transportation network is presented along with necessary data concerning ports and demand. The cost structures presented in chapter 2.3 and chapter 2.4 are applied to the case using appropriate regression models. Furthermore, an adapted version of the RSC method presented in chapter 3.2.3 is applied to the case. Finally, the results of the case study are presented.

The mathematical model presented in chapter 5 is written in Mosel code and solved in FICO Xpress IVE version 1.24.22. The epoch analysis is conducted by the MATLAB-Xpress interface, where Xpress is used to solve the mathematical TSP model and IP model. MATLAB is primarily used for prepossessing, route generation, storing iterative output data and plotting.

6.1 Case Description

A liner shipping company operating a conventional container feeder fleet considers to renew their fleet. Due to the increasing attention towards autonomy in other transportation segments, the company wants to consider autonomous fleets as an option. Since autonomous vessels are in an early stage of development, the owner wishes to include uncertain factors affecting the capital costs of autonomous vessels in the assessment. The shipping company is particularly interested in cost performance. Consequently, profit is neglected in this assessment. For the fleet to be viable the fleet must perform well, i.e. be cost efficient, in different contexts. In addition, the shipping company is interested in the optimal fleet configuration, i.e. the capacity of the vessels in the fleet, that minimizes cost.

Due to the uncertainty related to the cost of building autonomous vessels, the shipping company is curious about the impact autonomous technological developments have on production costs. In addition, the company is concerned about two economic factors that might affect operating costs and voyage costs, namely manning costs and oil price.

6.1.1 Regional Baltic Trade

The trade operated by the shipping company is based on data gathered from *liner-lib*, which is a benchmark suite for liner shipping network design (Løfstedt et al., 2010). Liner-lib consists of worldwide port and demand data, but the trade in this case study is restricted to a particular regional trade in the Baltic region. The trade consists of eight ports in total; seven spokes and one hub, or depot. The depot is located in Bremerhaven, Germany, which is the starting port for all vessels in the fleet. As discussed in chapter 3.1.2, this is a delivery problem, so the container cargo is shipped from the depot to the respective port of destination. The spoke ports have an individual weekly container demand as listed in the second last column in Table 6.1. The weekly demand originates from past data from Maersk Line which is from a given anonymous year, and has been subject to reasonable random perturbation and anonymisation to protect the confidentiality of Maersk Line data (Løfstedt et al., 2010). As stated by Løfstedt et al. (2010) the data does not represent a real business case, but it is encouraged to use the data in different settings to perform strategic analysis.

The ports are enumerated, and abbreviated according to their unique UNLO Code (United Nations Code for Trade and Transport Locations) as shown in the second column in Table 6.1. Liner-lib also provides port call costs for each individual port, as listed in the last column. The port call cost is added to the individual voyage costs for each port visit.

The distances between all of the ports are listed in Table 6.2. In this case study the distances are assumed to be *symmetrical*, meaning that the route distance to and from two ports are identical. The distances between the ports are extracted from the National Imagery and Mapping Agency (NIMA, 2001).

Table 6.1: Port data for the regional Baltic trade (adapted from Løfstedt et al., 2010).

#	UNLO	Name	Country	Type	Demand [TEU]	Port call [USD]
1	DEBRV	Bremerhaven	Germany	Depot	-	11,795
2	RUKGD	Kaliningrad	Russia	Port	536	1,062
3	FIKTK	Kotka	Finland	Port	374	1,182
4	FIRAU	Rauma	Finland	Port	36	18,552
5	NOAES	Ålesund	Norway	Port	20	24,098
6	NOBGO	Bergen	Norway	Port	34	17,435
7	NOKRS	Kristiansand	Norway	Port	12	24,076
8	NOSVG	Stavanger	Norway	Port	130	1,227

Table 6.2: Symmetric distance matrix given in nautical miles (Løfstedt et al., 2010).

	DEBRV	RUKGD	FIKTIK	FIRAU	NOAES	NOBGO	NOKRS	NOSVG
DEBRV	-	832	1,075	1,060	545	447	292	366
RUKGD	832	-	475	678	962	817	604	733
FIKTK	1,075	475	-	541	1,178	1,060	847	976
FIRAU	1,060	678	541	-	1,190	1,045	832	961
NOAES	545	962	1,178	1,190	-	167	393	278
NOBGO	447	817	1,060	1,045	167	-	226	111
NOKRS	292	604	847	832	393	226	-	142
NOSVG	366	733	976	961	278	111	142	-

The location of the numbered ports are illustrated in Figure 6.1 according to latitude and longitude coordinates provided by liner-lib. The hub port in Bremerhaven is indicated by a square, and the remaining spoke ports are indicated by circles. Weekly container demand given in TEUs is indicated by figures next to the spoke ports.

6.2 Cost Data

The cost data for the conventional and autonomous vessels was presented and discussed in Chapter 2. Operating costs, voyage costs and capital costs for the conventional fleet are calculated by a regression model as discussed in Chapter 2.3. Capital costs are solely determined by the capacity as input in the regression model, while operating costs and voyage costs also depend on the route distance determined in the multi-trip VRP model. The linear regression model for capital costs for the conventional fleet is shown in Figure 6.2. Initially, the goodness of fit corresponded to an R^2 of 0.44. To neglect the most extreme outliers from the regression model, the least absolute residuals (LAR) method in MATLAB was applied. Although the LAR method is applied, MATLAB does not remove the outliers from the regression plot. The goodness of fit increased accordingly, indicated by an R^2 of 0.97. The LAR method is ap-

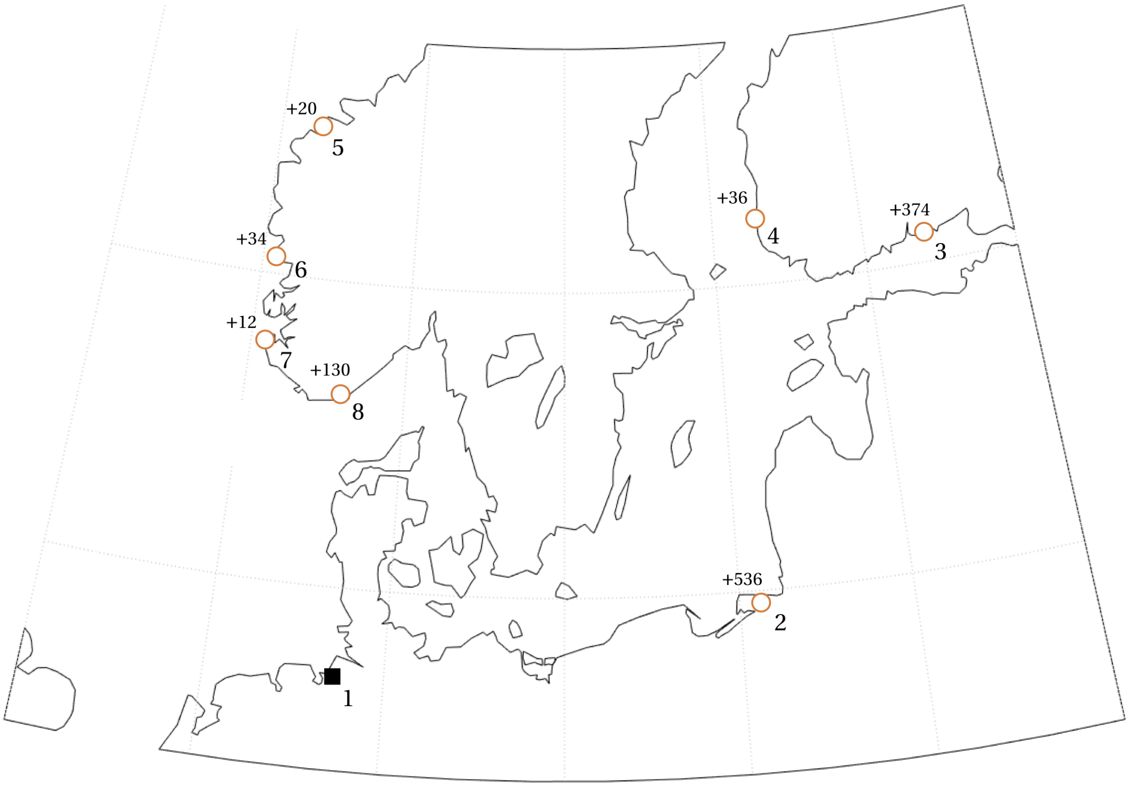

Figure 6.1: Actual location of the ports in the regional Baltic trade.

propriate when there are few outliers and when each data point is equally important, like in this case.

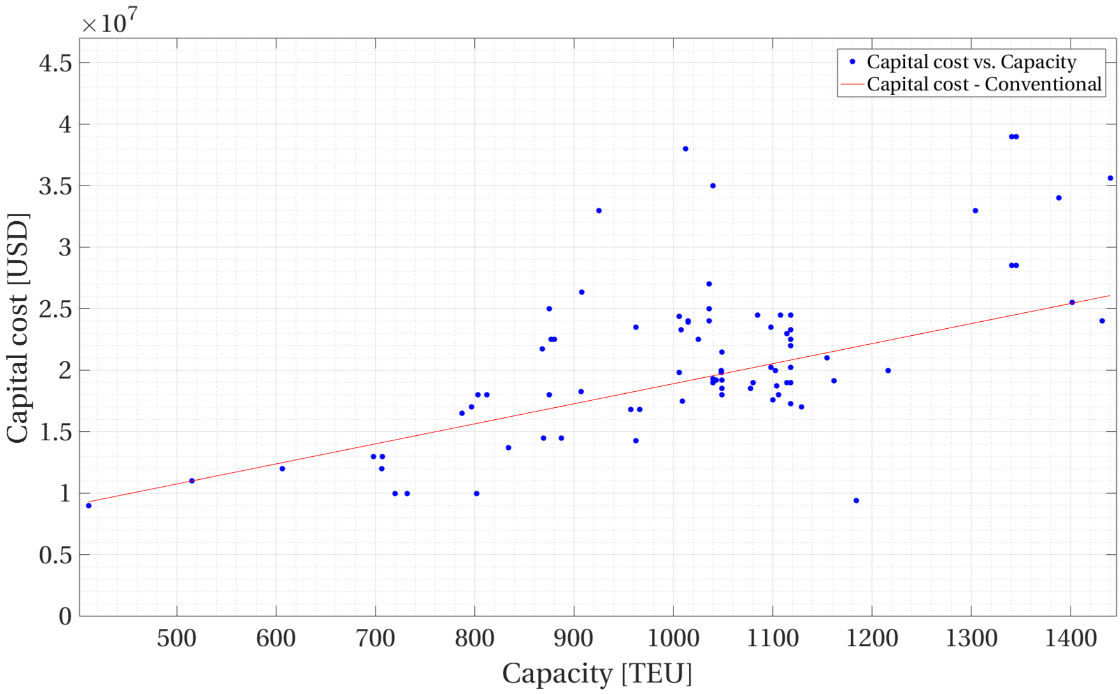

Figure 6.2: Capital cost regression model for the conventional fleet.

The regression models for the autonomous fleet are based on the regression models for the conventional fleets, but the models are adjusted according to the considered cost changes for autonomous vessels, as presented in Chapter 2.4.

6.3 Fleet Input Data

The multi-trip VRP model presented in chapter 5 is fed with fleet capacity data as listed in Table 6.3. The size of the input fleets are 2, 3 and 4 vessels. In order to investigate the effect of increasing fleet size the capacity range decreases when the fleet size is increased. The fleet list for a fleet size is determined by finding all unique combinations in the respective capacity range. For instance, for a fleet list of two vessels and capacity range $[300-1000]$ in increments of 50 TEUs, there are 105 unique fleet combinations. Some of these fleets are infeasible due to capacity constraints, which leads to a total number of 95 fleets, as shown in Table 6.3. The process is repeated for the different fleet sizes, which amounts to 382 fleet combinations. Since autonomous and conventional fleets are analyzed simultaneously, the total number of fleets considered amounts to $382 \times 2 = 762$.

Table 6.3: Initial fleet input data.

Fleet size	Capacity	Increment	Nr. of fleets
2	300 - 1000 TEU	50 TEU	95
3	100 - 700 TEU	50 TEU	202
4	100 - 550 TEU	50 TEU	84

6.4 Responsive Systems Comparison Method

Step 1 - Value-Driving Context Definition

The first step in the RSC method is to identify the problem statement, as already presented by the liner shipping company in the case description in chapter 6.1. The liner shipping company is interested in acquiring a fleet configuration that is cost efficient in different scenarios. To the fleet owner, this is achieved by minimizing capital costs, operating costs and voyage costs, and at the same time serving the weekly container demand in time.

Step 2 - Value-Driven Design Formulation

In the RSC method presented in chapter 3.2.3 the second step is to quantify attributes extracted from the value proposition in terms of utility functions. In this analysis utility functions are not used as a performance metric. Instead, capital cost is compared to operating costs and voyage costs in the tradespace.

Step 3 - Epoch Characterization

The epoch variables reflect the shipping company's expectations and concerns as presented in the case description. The epoch variables are shown in Table 6.4 and explained in detail in the following paragraphs.

Table 6.4: Epoch variables used in the case study.

Epoch Variable	Category	Scale	Unit	Range	Increment	Steps
Autonomous Development	Technological	Discrete	Condition	Conservative Moderate Developed	-	3
Bunker Index MGO	Economical	Discrete	[USD/ton]	500 - 750	125	3
Manning Costs	Economical	Discrete	[-]	0.9 - 1.1	0.1	3

Autonomous Development

The autonomous development variable reflect the stakeholder's concern regarding the uncertain building costs of autonomous vessels. As shown in Table 6.4 the autonomous development can be either conservative, moderate or developed. In the moderate state the capital cost of the autonomous vessel is equal to the estimate presented in chapter 2.4.3. This implies a capital cost increase of 7% compared to the capital costs of an conventional vessel. In a conservative scenario the capital cost is increased by 15%, similar to the conservative estimate conducted by Kretschmann et al. (2015). Similarly, in a developed scenario the capital cost is decreased by 15%.

Bunker Price

According to Løfstedt et al. (2010) the crude oil price is generally proportional to the price of bunker. The Bunker Index for marine gas oil (BIX MGO) is the average bunker price for a selection of global ports, and it provides an appropriate picture of the volatility of the bunker price. The BIX MGO from May 2015 to May 2018 is shown in Figure 6.3. The bunker price range is set from 500 USD/ton to 750 USD/ton in increments of 125 USD/ton, and has a direct impact on voyage costs. The price range is based on the minimum and maximum BIX from the past three years, as shown in Figure 6.3.

Figure 6.3: Bunker Index for MGO (BIX MGO) from May 2015 to May 2018. Source: www.bunkerindex.com

Manning Cost

Manning costs may constitute up to 50% of operating costs, depending on the size of the vessel, as discussed in chapter 2.4.1. In this analysis it is estimated that manning costs constitute 48% of operating costs in the feeder class. Manning costs are difficult to estimate since wages depend on the nationality of the crew, employment policies and the vessel's flag state (Stopford, 2009). The manning costs are either increased or decreased by 10% from the base case of the conventional vessel, which is equivalent to a 5% increase or decrease in operating costs. According to the evolution of the manning cost index presented by Drewry (2010), fluctuations of 10% are normal from year to year.

6.5 Results

Step 4-5: Tradespace Evaluation and Single Epoch Analysis

The three epoch variables lead to a total of 27 unique combinations of epochs. The base case scenario, or E_0, is characterized by a moderate autonomous development and bunker price, and a manning cost factor of 1, as summarized in Table 6.5. The tradespace for the base case scenario is depicted in Figure 6.4, where each point represents a particular fleet. The conventional and autonomous fleets are represented by circles and crosses, respectively. The different colors indicate the fleet size, as explained in Figure 6.4. The fleet configurations with minimum total costs are indicated by purple circles.

Table 6.5: Epoch configuration for the base case scenario.

Epoch	Autonomous Development	Bunker Index	Manning Costs
E_0	Moderate	625 USD/ton	1

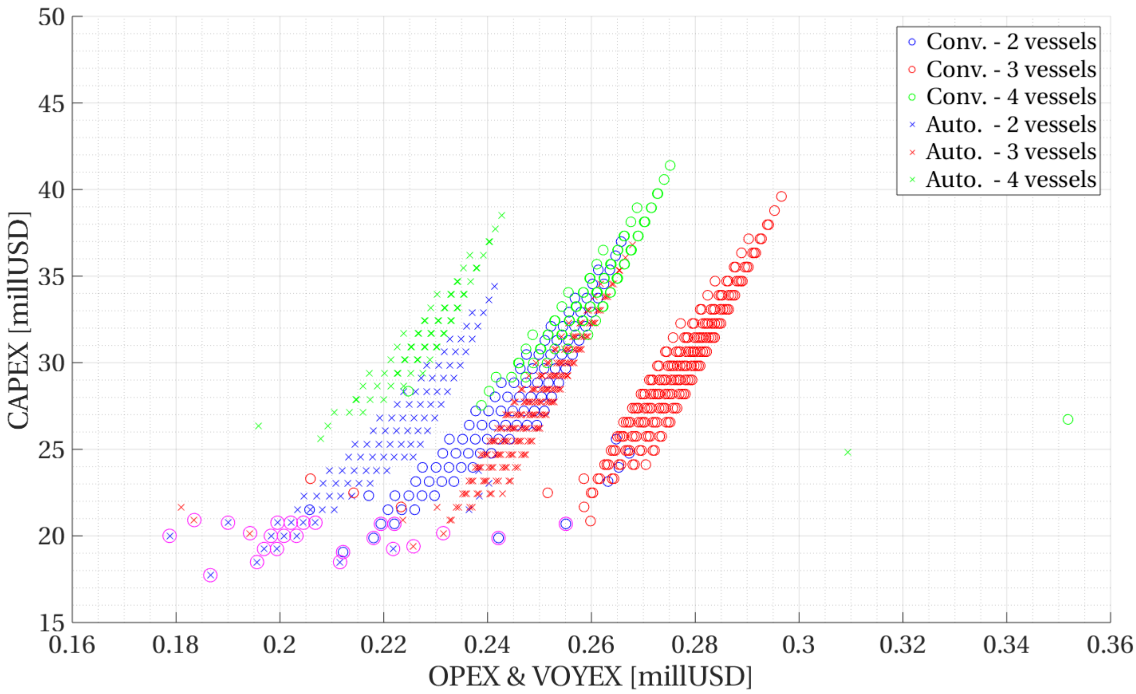

Figure 6.4: Tradespace for the base case scenario, denoted by E_0.

The majority of the low cost fleets are autonomous, as illustrated in Figure 6.4. Only six of the low cost fleets consist of conventional ships. The low cost fleets consist mainly of two vessels, while the remaining fleets consist of three vessels. Table 6.6 shows a selection of the fleets with minimal total cost in the base case scenario. Fleet 96 is the optimal fleet consisting of two autonomous vessels with capacity 300 TEU and 550 TEU. The total fleet cost, including capital costs, operational costs and voyage costs, amounts to USD 17.92 million.

A change in one of the epoch variables in Table 6.4 triggers the start of a new epoch. For instance, in the third epoch, E_3, the autonomous development has changed from moderate to conservative, while the bunker index and manning cost stay the same as in E_0. The epoch configuration for the third epoch is summarized in Table 6.7.

Table 6.6: Selection of fleets with minimal total cost in E_0.

Fleet ID	**Type**	**Capacities [TEU]**	**CAPEX [millUSD]**	**OPEX + VOYEX [millUSD]**	**Total [millUSD]**
96	Aut.	[300 550]	17.73	0.19	17.92
97	Aut.	[350 550]	18.48	0.21	18.69
106	Aut.	[300 600]	18.48	0.23	18.71
1	Conv.	[300 550]	19.10	0.21	19.31
116	Aut.	[300 750]	19.24	0.20	19.44

Table 6.7: Epoch configuration for the third epoch, E_3.

Epoch	Autonomous Development	Bunker Index	Manning Costs
E_3	Conservative	625 USD/ton	1

The tradespace for E_3 in Figure 6.5 shows that the autonomous fleets are now shifted upwards due to increased capital costs. The effect of the autonomous development is prominent, and now the top six of the fleets with minimal costs are conventional, as listed in Table 6.8.

Table 6.8: Selection of fleets with minimal total cost in E_3.

Fleet ID	**Type**	**Capacities [TEU]**	**CAPEX [millUSD]**	**OPEX + VOYEX [millUSD]**	**Total [millUSD]**
1	Conv.	[300 550]	19.10	0.21	19.31
11	Conv.	[350 550]	19.88	0.22	20.10
2	Conv.	[300 600]	19.88	0.24	20.12
21	Conv.	[400 550]	20.70	0.22	20.92
3	Conv.	[300 650]	20.70	0.26	20.96
191	Conv.	[100 150 550]	20.86	0.23	21.09
96	Aut.	[300 550]	21.28	0.19	21.47

The fleet with Fleet ID 1 is the best performing fleet in this epoch, with the same capacity configuration as the best performing fleet in the base case scenario. The autonomous fleet with minimum cost (Fleet ID 96) is USD 2.16 million more expensive than the best performing conventional fleet. Fleet 191 consists of three vessels, but it still manages to outperform the best performing autonomous vessel by USD 0.38 million.

Step 6: Multi-Epoch Analysis

The process is repeated for all possible epoch combinations, and the fleets that minimize cost in the individual epochs are traced. Figure 6.6 shows the frequency of oc-

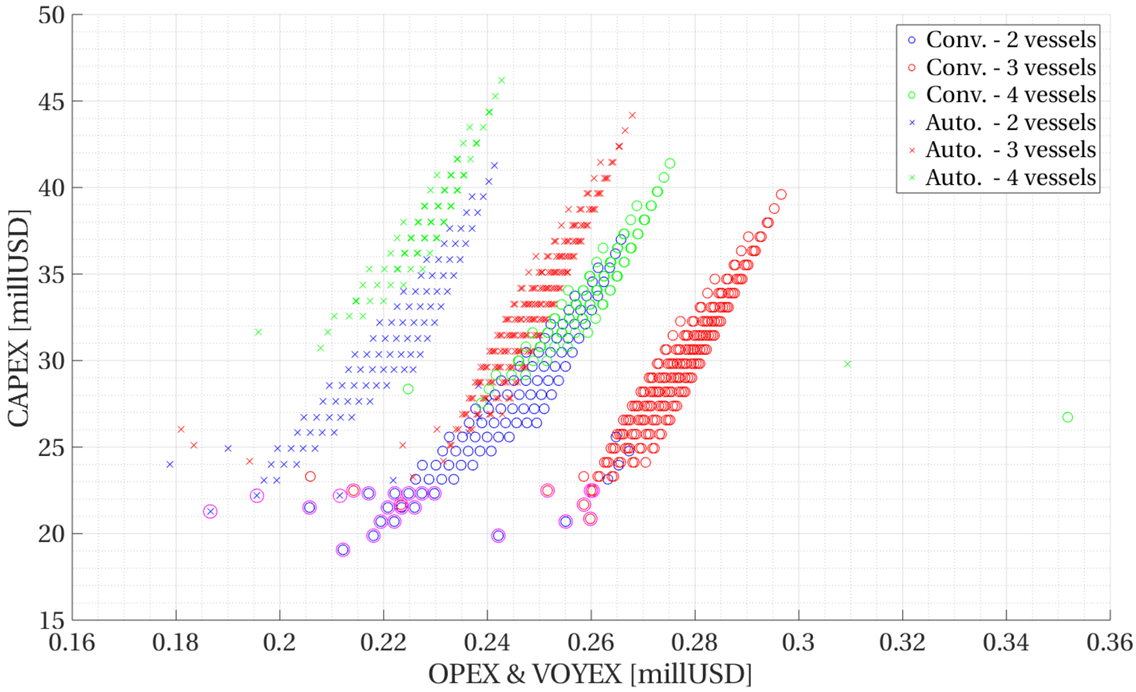

Figure 6.5: Tradespace for the third epoch, E_3.

currence of the optimal fleets in all possible epochs. The vertical axis represents the Fleet ID, while the horizontal axis represents the number of times the respective fleet is among the optimal fleets in an epoch. The autonomous fleets are marked in red and the conventional fleets are marked in blue.

In total there are three fleets that are among the optimal fleets in all 27 epoch combinations. As shown in Table 6.9, all of the optimal fleets are autonomous and consist of solely two vessels. The capacity configurations are relatively small having in mind that the capacity range in the input data (Table 6.3) was set to 300 - 1000 TEU for the fleets consisting of two vessels.

Table 6.9: Overall optimal fleet configurations from the optimal fleet trace.

Fleet ID	**Type**	**Capacities [TEU]**
96	Aut.	[300 550]
97	Aut.	[350 550]
106	Aut.	[300 600]

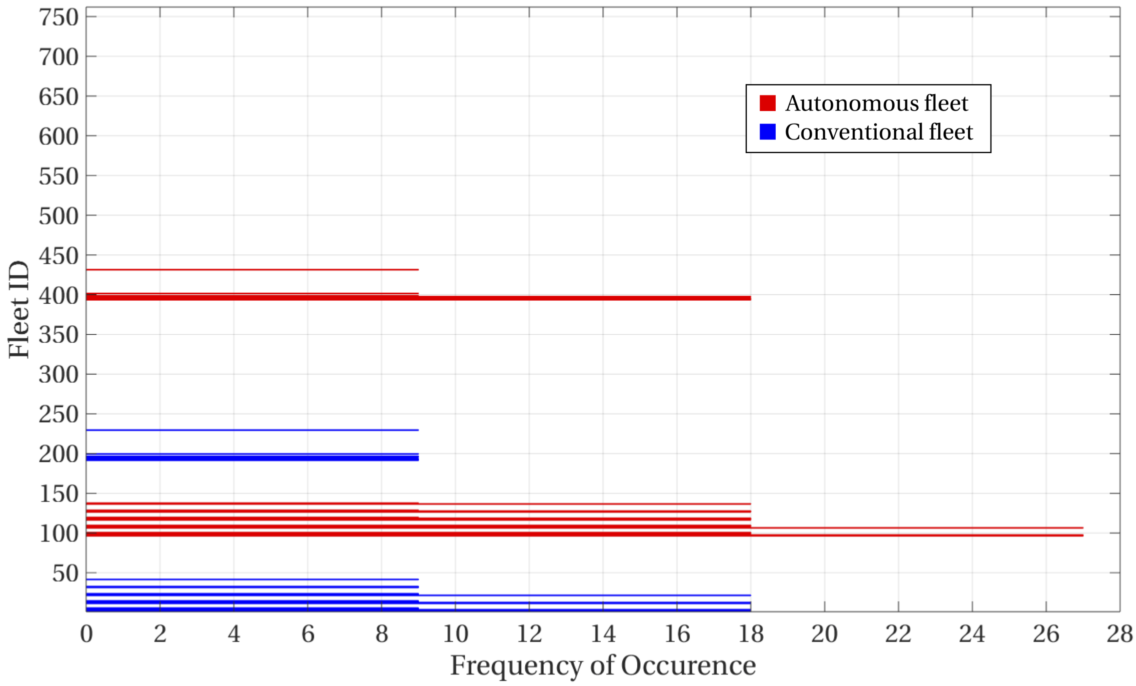

Figure 6.6: Optimal fleet trace - frequency of occurrence for all possible fleets.

Chapter 7

Discussion

In the first chapter of this thesis two research questions were stated. The aim of this chapter is to evaluate the results of the analysis, and to assess whether the research questions have been answered with the methods used. The first objective was to answer the research question: can the container shipping industry benefit of autonomous container feeder fleets? Next, what configuration of autonomous container feeder fleets minimizes cost and performs best in different contexts and needs?

Throughout this study, several indicators that the container shipping industry can benefit of autonomous vessel have been identified. The cost model for the autonomous vessel revealed a reduction in capital costs, operational costs and voyage costs compared to the conventional container vessel. The quantitative assessment of an autonomous dry bulk carrier conducted by the MUNIN project (Kretschmann et al., 2015), presented similar results with the exception of capital costs. The MUNIN project proposed an increase in capital costs of 10% compared to the conventional bulker, while the analysis in this thesis estimated a decrease of approximately 7.8%.

The fuel efficiency of the autonomous container vessel is estimated to decrease by approximately 6%, which is equal to the estimate proposed by the MUNIN project. It's plausible that this estimate is rather conservative as Arnsdorf (2014) proposes an estimate of potential fuel savings of 12-15%. As discussed in Chapter 2.4.2 the autonomous vessel in this analysis is restricted to only use MGO or higher grader fuels. This may be the reality since the required preparation process of HFO is difficult to automate (Willumsen, 2018). The use of higher grade fuels contribute to the greening of shipping.

The estimated cost changes for the autonomous vessel are associated with a high degree of uncertainty. First of all, the estimates are based on approximation methods meant for conventional vessels. The majority of the methods are based on historical data. It is conceivable that future innovative autonomous ship designs are so radically different from conventional designs that traditional approximation methods do not hold. For instance, innovative ships designs may lead to even more fuel efficient designs. The underlying cost data in the regressions models used in the case study originate solely from conventional manned vessels.

The required autonomous systems proposed in this thesis are primarily suggestions presented by the MUNIN project. The costs related to the technical systems are based on rough estimates. As stated by Kretschmann et al. (2017) autonomous ships may need additional systems and sensors onboard, which adds to the uncertainty of the cost assessment. The cost associated with the Shore Control Centre (SCC) is assumed to only affect the operating costs of the vessel; where the considered cost simply involve personnel cost at the SCC. It is not yet determined who is going to manage the SCC or pay for the acquisition. In a real world scenario the liner shipping company may be the owner and operator of the SCC, which in turn will contribute to increase costs.

The biggest challenge in the assessment of costs has been the very limited access to precise data. It has been difficult to obtain results with practical consequences when the underlying predictions are at a high level of uncertainty. However, it is shown in the cost assessment that autonomous container vessels have the potential to reduce costs. The final answer rely on the cost of required technology, and the fuel efficiency and newbuilding price of a potential new ship design. Future development of autonomous ship designs and concepts will eventually lead to greater insight in required systems, and hopefully provide more detailed cost data.

In this analysis the first six steps of the Responsive Systems Comparison (RSC) method was applied to the case study. The results from the final step, the multi-epoch analysis, showed that mainly three fleets consistently were among the fleets with minimal costs in each epoch. As shown in Table 6.9 all of the optimal fleets are autonomous.

Kretschmann et al. (2017) conclude that the autonomous dry bulk carrier is less expensive only if it is more fuel efficient than the conventional vessel. However, if the autonomous bulker uses marine diesel oil (MDO) while the reference bulker still uses HFO, the autonomous bulker is far more expensive. This is the opposite of the results in the analysis in this thesis; the cost of the optimal autonomous fleet in the base case scenario is USD 1.39 million less than the optimal conventional fleet. Referring to Table 6.6 the autonomous fleet (Fleet ID 96) and the conventional fleet (Fleet ID 1) have the same capacity configuration of 300 TEU and 500 TEU, which allows for the fleets to be compared. It's interesting that the autonomous fleet minimizes cost although the container ships use MGO as main fuel.

The most challenging part of the analysis was to capture the difference between the autonomous and conventional fleets using the RSC approach. It turns out that the results from the RSC method are rather deterministic and self-evident. With the choice of epoch variables and CAPEX as performance measure, the multi-epoch analysis (MEA) acts more like a sensitivity analysis. The tradespace exploration loses its main purpose when CAPEX is used instead of utility. It is difficult for the RSC practitioner to evaluate the "trade-off" between CAPEX, and OPEX and VOYEX. Well-defined utility functions were difficult to formulate due to the conceptual nature of autonomous shipping, especially when the aim is the compare two systems with different technical requirements.

Against this background, it is not likely that the results from the MEA have significant practical consequences. The uncertainty associated with costs and the lack of randomness makes it difficult to compare fleets in different contexts in the MEA. However, the analysis conducted in this thesis may serve as a suitable framework for further consideration. When autonomous shipping is more developed, it is easier to map the stakeholder's needs and consequently define robust utility functions.

The RSC method combined with a multi-trip VRP is an interesting evaluation approach because both dynamic and static aspects are considered. With well-defined utility functions and epoch variables that affect the routing decisions in every epoch, the approach can be a powerful tool for a potential ship investor. The results will influence both strategic and tactical decisions. The key challenge is to delimit the routing model and find the right balance between complexity and simplification of the real world problem. Gaspar et al. (2012) are successful in the use of a similar approach to handle contextual uncertainty in the design of an anchor handling tug supply vessel. The approach combines EEA and Ship Design and Deployment Problem (SDDP) to determine the performance of the vessel in different epochs. The key difference lies in the use of well-defined utility functions and contextual routing decisions to produce useful results.

Chapter 8

Conclusion

8.1 Concluding Remarks

This thesis compares conventional and autonomous container feeder fleets in terms of costs, taking contextual uncertainties into account. The costs changes that occur for autonomous container vessels are estimated using approximation methods with a conventional container vessel as reference. The required infrastructure and technical systems for autonomous operation are identified by investigating existing research projects on the topic. A model for a multi-trip Vehicle Routing Problem was developed to obtain operational costs and voyage costs for a set of conventional and autonomous fleets. The model was solved using a route generation algorithm and an IP model. The fleets were compared in different contexts by applying the Responsive Systems Comparison method for Multi-Epoch Analysis. The output of the model served as the input for the epoch analysis; following an iterative pipeline procedure.

It's safe to argue that the shipping industry can benefit from autonomous container feeder fleets on several levels. Autonomous shipping has the potential to reduce costs and contribute to the greening of shipping. New innovative autonomous ship designs may further increase fuel efficiency, and autonomy opens up the door for alternative fuels. This thesis shows that autonomous container fleets have lower costs than conventional container fleets in a variety of different contexts. However, the results from the analysis are related with a high degree of uncertainty. The uncertainties are primarily due to the conceptual state of autonomy in container shipping, and limited access to precise cost data for autonomous vessels.

The results from the case study revealed that the optimal configuration of an autonomous fleet in the regional Baltic trade consisted of two vessels with capacity of 300 TEU and 550 TEU. This result is case-specific and should not be interpreted as a general result. However, the approach can easily be adapted to a different case, and the model can be adjusted accordingly.

The results from the contextual analysis have minimal impact on the development of autonomous ships in liner shipping. The way the epoch analysis was applied to the case problem failed to give results of practical consequence. However, the overall approach can serve as a powerful tool for further research when autonomous ships are more developed.

8.2 Recommendations for Further Work

The RSC method combined with routing is an interesting approach that is encouraged to use in further research of autonomous vessels. However, more detailed cost data and vessels specifications should be available to achieve interesting results. Future physical autonomous ship designs would hopefully provide insight to more detailed fuel consumption estimates and production costs. In a developed scenario it may also be more convenient to formulate well-defined utility functions from the stakeholder's needs. The approach can then potentially be used as a design tool in a consulting process.

One of the benefits of innovative autonomous ship designs is the additional space for cargo. It would be interesting to investigate the potential impact of increased revenue due additional cargo space. The analysis in this thesis is restricted to liner shipping, but other shipping modes may also benefit from autonomy. In tramp mode, fleets of fully autonomous vessels with smaller capacity could potentially be used in a point-to-point network structure.

Bibliography

ABS (2014). *Ship Energy Efficiency Measures*. American Bureau of Shipping. URL: https://ww2.eagle.org/content/dam/eagle/advisories-and-debriefs/ABS_Energy_Efficiency_Advisory.pdf.

Amdahl, Jørgen, Anders Endal, Geir Fuglerud, Liv Randi Hultgreen, Knut Minsaas, Magnus Rasmussen, Bjørn Sillerud, Bjørn Sortland, Sverre Steen, and Harald Valland (2014). *TMR4105 - Marin Teknikk Grunnlag*. Trondheim: Kompendieforlaget.

Arnsdorf, Isaac (2014). *Rolls-Royce Drone Ships Challenge $375 Billion Industry: Freight*. URL: https://www.bloomberg.com/news/articles/2014-02-25/rolls-royce-drone-ships-challenge-375-billion-industry-freight.

Bendall, H.B and A.F Stent (2001). "A Scheduling Model for a High Speed Container Service: A Hub and Spoke Short-Sea Application". In: *International Journal of Maritime Economics* 3 (3), pp. 262–277.

Bertram, Volker (2016). *Unmanned & Autonomous Shipping - A Technology Review*. 10th Symposium on High-Performance Marine Vehicles, pp. 10–24.

Bertram, Volker and Herbert Schneekluth (1998). *Ship Design for Efficiency and Economy*. 2nd ed. Elsevier. ISBN: 978-0-7506-4133-3.

Blanding, H. C. (1987). *Automation of Ships and the Human Factor*. Philadelphia: SNAME Ship Technology and Research Symposium.

Blendermann, Werner (1996). *Wind Loading of Ships - Collected Data from Wind Tunnel Tests in Uniform Flow*. Bericht Institut für Schiffbau.

Brouer, Berit D., Christian Vad Karsten, and David Pisinger (2016). *Big Data Optimization: Recent Developments and Challenges*. Springer International Publishing Switzerland. Chap. Big Data Optimization in Maritime Logistics, pp. 319–343.

Burmeister, Hans-Christoph, Wilko C. Bruhn, Ornulf J. Rodseth, and Thomas Porathe (2014). "Can Unmanned Ships Improve Navigational Safety?" In: Transport Research Arena. Paris.

Carlson, J. M. and J. Doyle (2000). "Highly Optimized Tolerance: Robustness and Design in Complex Systems". In: *Physical Review Letters* 84 (11), pp. 2529–2532.

Chen, Chao and Qingcheng Zeng (2010). “Designing Container Shipping Network Under Changing Demand and Freight Rates”. In: *Transport* 25 (1), pp. 46–57.

Christiansen, Marielle, Kjetil Fagerholt, Bjorn Nygreen, and David Ronen (2012). *Ship Routing and Scheduling in the New Millenium*. Elsevier.

Christiansen, Marielle, Kjetil Fagerholt, Bjørn Nygreen, and David Ronen (2007). *Handbook in Operations Research and Management Science*. Vol. 14. Elsevier. Chap. 4. Maritime Transportation, pp. 189–284.

Christiansen, Marielle, Kjetil Fagerholt, and David Ronen (2004). *Ship Routing and Scheduling: Status and Perspectives*. Vol. 38. Transportation Science, pp. 1–18.

Damen (2017). *Container Feeder 800 Product Sheet*. Damen Shipyards Group. URL: https://products.damen.com/-/media/Products/Images/Clusters-groups/Shipping/Container-Feeder/CFe-800/Documents/Product_Sheet_Damen_Container_Feeder_800_06_2017.pdf.

Drewry (2010). *Ship Operating Costs Annual Review and Forecast. Annual Report 2010/11*. Drewry Publishing, Nigel Gardiner.

Fagerholt, Kjetil (1999). “Optimal Fleet Design in a Ship Routing Problem”. In: *International Transactions in Operational Research* 6 (5), pp. 453–464.

— (2004). *Designing optimal routes in a liner shipping problem*. Vol. 31. 4. Maritime Policy & Management, pp. 259–268.

Fagerholt, Kjetil, Marielle Christiansen, Bjørn Nygreen, and David Ronen (2012). *Ship Routing and Scheduling in the New Millenium*. Elsevier.

Gaspar, Henrique M., Stein Ove Erikstad, and Adam M. Ross (2012). “Handling Temporal Complexity in the Design of Non-Transport Ships Using Epoch-Era Analysis”. In: *The International Journal of Maritime Engineering*.

Hillier, Frederick S. and Gerald J. Lieberman (2001). *Introduction to Operations Research*. New York: McGraw-Hill.

Kongsberg (2017). “Autonomous Ship Project, Key Facts about YARA Birkeland”. In: URL: https://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/4B8113B707A50A4FC125811D00407045?OpenDocument.

Kretschmann, Lutz, Hans-Christoph Burmeister, and Carlos Jahn (2017). *Analyzing the Economic Benefit of Unmanned Autonomous Ships. An Exploratory Cost-comparison between an Autonomous and a Conventional Bulk Carrier*. Elsevier.

Kretschmann, Lutz, Ørnulf Rødseth, Bénédicte Sage Fuller, Helen Noble, Jeffrey Horahan, and Hugh McDowell (2015). *Quantitative Assessment. MUNIN Deliverable 9.3*. MUNIN Report.

Kristensen, Hans Otto and Marie Lützen (2013). “Prediction of Resistance and Propulsion Power of Ships”. In: 2.4.

Løfstedt, B., J. F. Alvarez, C. E. M. Plum, D. Pisinger, and M. M. Sigurd (2010). *An Integer Programming Model and Benchmark Suite for Liner Shipping Network Design.* DTU Management Engineering Report 19. Kgs. Lyngby: DTU Management.

Lun, Y.H Venus and Michael Browne (2009). “Fleet Mix in Container Shipping Operations”. In: *International Journal of Shipping and Transport Logistics* 1 (2), pp. 103–118.

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

MacKinnon, Scott N., Yemao Man, and Michael Baldauf (2015). *Final Report: Shore Control Centre. MUNIN Deliverable 8.8.* MUNIN Report.

MAN (2014). *Propulsion Trends in Container Vessels.* MAN Diesel & Turbo. URL: https://marine.mandieselturbo.com/docs/librariesprovider6/technical-papers/propulsion-trends-in-container-vessels.pdf?sfvrsn=20.

Marquard & Bahls (2015). *Marine Gasoil.* URL: https://www.marquard-bahls.com/en/news-info/glossary/detail/term/marine-gasoil-mgo.html.

Mau, G. (1984). *Handbuch Dieselmotoren im Kraftwerks- und Schiffsbetrieb.* Wiesbaden: Vieweg & Teubner.

MUNIN (2016). *Research in Maritime Autonomous Systems. Project Results and Technology Potentials.* URL: www.unmanned-ship.org/munin/news-information/downloads-information-material/munin-dissemination/ (visited on 05/05/2018).

NIMA (2001). *Distances Between Ports.* National Imagery and Mapping Agency. URL: http://www.marcon.com/library/resources/DBP/DBP.html. Pub. 151.

Norstad, Inge (2017). “Lecture Notes. TMR8 Fleet Scheduling and Supply Chains”. NTNU.

Pettersen, Sigurd S., Carl F. Rehn, Jose J. Garcia, Stein O. Erikstad, Per O. Brett, Bjørn E. Asbjørnslett, Adam M. Ross, and Donna H. Rhodes (2017). “Ill-Structured Commercial Ship Design Problems: The Responsive Systems Comparison Method on an Offshore Vessel Case”. In: *Journal of Ship Production and Design* 34 (1).

Porathe, Thomas, Nicole Costa, and Åsmund Tjora (2014). *Organizational Lay-out of SOC. MUNIN Deliverable 7.4.* MUNIN Report.

Rodseth, Ornulf and Hans-Christoph Burmeister (2012). *Developments Toward the Unmanned Ship.* Hamburg, Germany: Proceedings of International Symposium Information Ships.

Ronen, David (1983). *Cargo Ships Routing and Scheduling: Survey of Models and Problems.* School of Business Administration, University of Missouri.

Ronen, David, Kjetil Fagerholt, and Marielle Christiansen (2004). In: *Transportation Science* 38 (1): *Ship Routing and Scheduling: Status and Perspectives*, pp. 1–18. DOI: 10.1287/trsc.1030.0036.

Ross, Adam M. (2006). "Managing Unarticulated Value: Changeability in Multi-Attribute Tradespace Exploration". PhD thesis. Massachusetts Institute of Technology.

Ross, Adam M. and Daniel E. Hastings (2005). *The Tradespace Exploration Paradigm.* Cambridge: INCOSE.

— (2006). *Assessing Changeability in Aerospace Systems Architecting and Design Using Dynamic Multi-Attribute Tradespace Explorations.* American Institute of Aeronautics and Astronautics.

Ross, Adam M., Hugh L. N. McManus, Donna H. Rhodes, Daniel E. Hastings, and Andrew Long (2009). *Responsive Systems Comparison Method: Dynamic Insights into Designing a Satellite Radar System.* Paper presented at the AIAA SPACE 2009 Conference & Exposition, Pasadena.

Responsive Systems Comparison Method: Case Study in Assessing Future Designs in the Presence of Change (2008a). (AIAA Space 2008). San Diego, California.

Ross, Adam M. and Donna H. Rhodes (2008b). *Using Natural Value-Centric Time Scales for Conceptualizing System Timelines through Epoch-Era Analysis.* INCOSE.

Rothblum, A. M. (2000). *Human Error and Marine Safety.* US Coast Guard Research & Development Center.

Sanquist, Thomas F. (1992). *Human Factors in Maritime Applications: A New Opportunity for Multi-Modal Transportation Research.* In Proceedings of the 36th Human Factors and Ergonomics Society Annual Meeting. SAGE Publications, pp. 1123–1127.

Saxton, Brita (2016). "Fuels in the Baltic Sea after SECA". In: URL: https://www.trafa.se/globalassets/pm/2016/pm-2016_12-fuels-in-the-baltic-sea-after-seca.pdf.

A Method for Selecting Affordable System Concepts: A Case Application to Naval Ship Design (2014). (Conference on Systems Engineering Research).

Sigurd, Mikkel M., Nina L. Ulstein, Bjørn Nygreen, and David M. Ryan (2005). *Column Generation.* Boston, MA: Springer. Chap. Ship Scheduling with Recurring Visits and Visit Separation Requirements, pp. 222–245. ISBN: 978-0-387-25486-9.

Stensvold, T. (2017a). "Rolls-Royce vil bygge containerskip etter Lego-prinsippet". In: URL: https://www.tu.no/artikler/rolls-royce-vil-bygge-containerskip-etter-lego-prinsippet/377376.

— (2017b). "Skal være klar til bruk i 2020: I dag startet testingen av verdens første autonome containerskip". In: URL: https://www.tu.no/artikler/skal-vaere-klar-til-bruk-i-2020-na-testes-verdens-forste-autonome-containerskip/408426.

Stopford, M. (2009). *Maritime Economics*. Routledge.

Waterborne TP (2011). *Waterborne Implementation Plan: Issue May 2011*. Waterborne Strategic Research Agenda.

Willumsen, Torgeir (2018). *A Commercial Reality Check for Autonomous Shipping in 2018*. URL: http://www.seatrade-maritime.com/news/europe/a-commercial-reality-check-for-autonomous-shipping-in-2018.html.

Wilson, Jonathan (2017). *Rolls-Royce Marine unveils 'Electric Blue' modular smart shipping concept*. URL: https://eandt.theiet.org/content/articles/2017/02/rolls-royce-marine-unveils-electric-blue-modular-smart-shipping-concept/ (visited on 12/05/2018).

Appendix A

List of Abbreviations

AEMCS	Autonomous Engine Monitoring and Control System
ASM	Autonomous Sensor Module
BHP	Brake Horse Power
BIX	Bunker Index
CAPEX	Capital Expenditure
EEA	Epoch-Era Analysis
EPV	Expected Present Vale
HFO	Heavy Fuel Oil
IMO	International Maritime Organization
IP	Integer Programming
MARPOL	Marine Pollution
MEA	Multi-Epoch Analysis
MDO	Marine Diesel Oil
MGO	Marine Gas Oil
MTTP	Maritime Transportation Planning Problems
OOW	Officer on Watch
OPEX	Operating Expenditure
RMSS	Remote Maneuvering Support System
SCC	Shore Control Centre
SDDP	Ship Design and Deployment Problem
TSP	Travelling Salesman Problem

VOYEX Voyage Expenditure

VRP Vehicle Routing Problem

VTS Vessel Traffic Servce

Appendix B

MATLAB Code

B.1 Fleet Generation

```
%= Written by Harald Ulvestad Salvesen, Spring 2018
%= Norwegian University of Science and Technology

%Load network of nodes (ports) and distances between each node
graphPlot2;

%Determine number of ports in transportation network
nNodes = size(FullMap.Nodes.NodeID,1);

%Total number of vessels in a fleet
nVessels = 2;

%Load regression analysis
getReg;

%Bunker Price
bunkerPrice = 641; %[$/tons]

%Port demands [TEUs]
Demand = FullMap.Nodes.Demand(2:end)';

%Capacity range [TEUs] - set range
CapacityRange = [300:50:1000];
```

```
Steps = size(CapacityRange,2);
cCap = combntns(CapacityRange,nVessels);

%Delete fleets that can't ship the largest demand of 536 TEUs
pp = 1;
for cc = 1:size(cCap,1)
   if cCap(cc,2) > 536
      combsCap(pp,:) = cCap(cc,:);
      pp = pp + 1;
   end
end

%Determine number of iterations (number of fleets)
nIterations = size(combsCap,1);
for ts = 1:nIterations
Capacity = combsCap(ts,:);

%Load routegeneration for specified fleet
RouteGen();

    t = 1;
    for pp = 1:nVessels

        nRoutes = size(FeasibleRoutes{1,pp},1);
        for jjj = 1:nRoutes
            SelectRoute = FeasibleRoutes{1,pp};
            Ports = SelectRoute(jjj,:);

            %Delete "zero-ports"
            D = find(Ports > 0);

            clear Delta;
            clear demandNodes;
            %Identify the NodeID based on demand from RouteGen.m
            for i = 1:D(end)
                demandNodes(1) = 1;
                demandNodes(i+1) = .....
                    FullMap.Nodes.NodeID(find(FullMap.Nodes.Demand...
                    == Ports(i)));
            end

            for ii = 1:size(demandNodes,2)
```

```
        remNodes = find(demandNodes ~= demandNodes(ii));
        remNodes = demandNodes(remNodes);
        conEdges = edgeList{2,demandNodes(ii)};

        for jj = 1:size(remNodes,2)
            conNodes(jj) = ...
                find(edgeList{1,demandNodes(ii)} ==.....
                 remNodes(jj));
        end

        for jj = 1:size(conNodes,2)
            Delta(ii,jj) = conEdges(conNodes(jj));
        end

    end

    Distance = FullMap.Edges.Distance';
    nDist = size(Distance, 2);
    EdgeSet = unique(Delta)';
    nEdgeSet = size(EdgeSet,2);
    nEdge = size(Delta,2);

    %Crate E and W for subtour constraints:
    i = 1;
    for gg = 2:size(demandNodes,2)
        utfall{i,1} = nchoosek(demandNodes,gg);
        node = utfall{i,1};
        [rows cols] = size(node);
        mm = 1;
        for k = 1:rows
            for s = 2:cols
                nodeStart = node(k,1);
                nodeList = edgeList{1,nodeStart};
                loc = find(nodeList == node(k,s));
                edgeTempList = edgeList{2,nodeStart};
                edge(mm) = edgeTempList(loc);
                W{i,1} = edge';
                mm = mm + 1;
            end
        end
        clear edge
        i = i + 1;
    end
```

```
Wsize = size(W,1);
for j = 2:Wsize
    if j < 6
        edge = W{j,1};
        edge = vec2mat(edge,j);
        W{j,1} = edge;
    else
        edge = W{j,1};
        W{j,1} = edge';
    end
end

%Determine dimension of each entry of W
for i = 1:Wsize
    [nrW(i) ncW(i)] = size(W{i,1});
end

%Write input data to TSP model
writeFleetTSP();

%Run TSP model (Xpress IVE)
moselexec('TSPmod3.mos');
clear W
clear edge
clear utfall

%Save optimal routes and distance to array in ...
   MATLAB workspace
for nn = 1:numel(FullMap.Nodes.NodeID)
    comp = find(demandNodes == ...
       FullMap.Nodes.NodeID(nn));
    T = isempty(comp);
    if T == 1
        nodeTSP(nn) = 0;
    else
        nodeTSP(nn) = 1;
    end
end
distTSP(jjj) = objval;
%Save opt. distances to MATLAB
solTSP{1,pp} = distTSP';
flowTSP{jjj} = nodeTSP';
%Save opt. routes(flow) to MATLAB
```

```
            solTSP{2,pp} = flowTSP';
        end
        clear flowTSP; clear distTSP; clear nodeTSP;
    end
%VRP - Parse candidate routes and corresponding route cost to ...
   Xpress IVE
%Determine total of number of routes for this particular instance
countRoute = 0;
for v = 1:nVessels
    countRoute = countRoute + numel(solTSP{1,v});
end

%Write fleet data to VRPData.txt for interpretation in Xpress ...
   IVE;
writeFleetVRPtest2();

%Run VRP-model (Xpress) from MATLAB
moselexec('VRPmodTIME.mos');

%Structure TSP data to fit VRP data
for v = 1:nVessels
    solTSP{1,v} = solTSP{1,v}';
    A{v} = solTSP{1,v};
end

cntr = 1;
for v = 1:nVessels
    nR = numel(solTSP{2,v});
    for bb = 1:nR
        B{cntr} = solTSP{2,v}{bb};
        cntr = cntr + 1;
    end
end
B = B';

 distTSPVRP = cell2mat(A)';
 vrpflow = vrpflow';

%Interpret solutions from Xpress (VRP model) - find ...
   individual cost
for v = 1:nVessels
    optRoutes = find(vrpflow(:,v) == 1);
    vesselDist = distTSPVRP(optRoutes);
```

```
    vesselDist = sum(vesselDist);
    routingSol{ts,v} = B(optRoutes);

    %Extract port call cost
    nOR = numel(optRoutes);
    for pc = 1:nOR
        OR = B{optRoutes(nOR)};
        visitPorts = find(OR == 1);
        PortCall = FullMap.Nodes.PortCall(visitPorts);
        PortCall = sum(PortCall);
    end

    %Determine design speed for each vessel
    dSpeed = Speed_reg(Capacity(v)); %[kn = nm/h]
    %Determine sailing time
    timeAtSea = (vesselDist/dSpeed); %[hours]
    %Calculate deisgn fuel cost
    fuelCost = ...
       FC_Conv_Reg(Capacity(v))*timeAtSea*bunkerPrice; %[tons]
    %Calculate OPEX
    OPEX = RegOPEX(Capacity(v))*timeAtSea*24;

    Cost(v) = fuelCost + OPEX + PortCall;

    %Calculate CAPEX
    CAPEX(v) = NB_Price_Reg(Capacity(v));

    %Save route distance
    resultDist(ts,v) = vesselDist;
    resultPC(ts,v) = PortCall;

clear optRoutes;
end
Cost = sum(Cost);
CAPEX = sum(CAPEX);
Cost = Cost/10^6;
tsCost(ts) = Cost;
tsCAPEX(ts) = CAPEX;
clear Cost; clear CAPEX;
end


%Plot results
```

```
sz=13;
scatter(tsCost,tsCAPEX,sz,'blue')
grid on
grid minor
xlabel('OPEX & VOYEX [millUSD]')
ylabel('CAPEX [millUSD]')
xlim([0 max(tsCost)+0.02]);
ylim([0 max(tsCAPEX)+1]);
```

B.2 Route Generation

```
%==================================================
% RouteGen.m - Route generating script
%==================================================
% Algorithm description:
%  - All feasible routes with respect to capacity
%    constraints are generated
%==================================================
%= Written by Harald Ulvestad Salvesen, Spring 2018
%= Norwegian University of Science and Technology
%==================================================

%clear all;clc;
[e numPorts] = size(Demand);

%Generate all route combinations
%1 port-trip
Routes = [Demand'];
Routes(:,2:numPorts) = zeros;

%2 port-trip
twoPort = combntns(Demand,2);
[twoPortSize n] = size(twoPort);
Routes(numPorts+1:numPorts+twoPortSize,:) = zeros;
Routes(numPorts+1:numPorts+twoPortSize,1:n) = twoPort;

%3 port-trip
threePort = combntns(Demand,3);
[threePortSize n] = size(threePort);
[last h] = size(Routes);
Routes(last+1:last+threePortSize,:) = zeros;
Routes(last+1:last+threePortSize,1:n) = threePort;

%4 port-trip
fourPort = combntns(Demand,4);
[fourPortSize n] = size(fourPort);
[last h] = size(Routes);
Routes(last+1:last+fourPortSize,:) = zeros;
Routes(last+1:last+fourPortSize,1:n) = fourPort;

%5 port-trip
```

```
fivePort = combntns(Demand,5);
[fivePortSize n] = size(fivePort);
[last h] = size(Routes);
Routes(last+1:last+fivePortSize,:) = zeros;
Routes(last+1:last+fivePortSize,1:n) = fivePort;

%6 port-trip
sixPort = combntns(Demand,6);
[sixPortSize n] = size(sixPort);
[last h] = size(Routes);
Routes(last+1:last+sixPortSize,:) = zeros;
Routes(last+1:last+sixPortSize,1:n) = sixPort;

%7 port-trip
[last h] = size(Routes);
Routes(last+1,1:numPorts) = Demand;

%Calculate total route load
[last h] = size(Routes);
for i = 1:last
    Routes(i,numPorts+1) = sum(Routes(i,1:6));
end

%Find all possible routes where the sum is <= Capacity(i)

for qq = 1:nVessels

    for k = 1:last
        if Routes(k,numPorts+1) <= Capacity(qq)
            Cand(k,:) = Routes(k,1:numPorts);
        end
    end
    %Delete zero-rows
    Cand( ~any(Cand,2), : ) = [];
    %Extract feasible routes for vehicle qq to a cell array
    FeasibleRoutes{qq} = Cand;
end
```

B.3 Regression Models

```
%==========================================================
%= getReg.m - Regression models
%= Written by Harald Ulvestad Salvesen, Spring 2018
%= Norwegian University of Science and Technology
%==========================================================

% %Import and structure regression data
 regRead = csvread('regdata3.csv');

 TEUClassAll = regRead(1,:);

 OpEx = regRead(4,:); % [$/hour]
 TEUClassStop = regRead(5,:);
 RegOPEX = fit(TEUClassAll', OpEx', 'poly1');

% Conventional Fleet
%==========================================================
%Newbuilding price
%==========================================================
%Read newbuilding price data
num = xlsread('nb.xlsx');

TEUclass = num(:,1);
NB_Price = num(:,2);
[xData, yData] = prepareCurveData( TEUclass, NB_Price );

% Set up fittype and options
ft = fittype( 'poly1' );
opts = fitoptions( 'Method', 'LinearLeastSquares' );
opts.Robust = 'LAR';

% Fit model to data.
[NB_Price_Reg, gof] = fit(xData, yData, ft, opts );

% Plot fit with data.
figure( 'Name', 'Newbudiling Price - Conventional');
h = plot(NB_Price_Reg, xData, yData );
legend( h, 'Newbuilding price vs. Capacity', ...
'Newbudiling Price - Conventional', 'Location', 'NorthEast' );
```

```
%Label axes
xlabel('Capacity [TEU]')
ylabel('Newbuilding price [USD]')
grid on
grid minor
set(gca,'FontName','Utopia');
set(gca,'FontSize', 16);

%===========================================================
%Fuel consumption
%===========================================================

FC = xlsread('seaweb.xls',2,'A1:B92');

TEUclass = FC(:,1);
FC_conv = FC(:,2);

[xData, yData] = prepareCurveData( TEUclass, FC_conv );

% Set up fittype and options
ft = fittype( 'poly1' );
opts = fitoptions( 'Method', 'LinearLeastSquares' );
opts.Robust = 'LAR';

% Fit model to data.
[FC_Conv_Reg, gof] = fit(xData, yData, ft, opts );

% Plot fit with data.
figure( 'Name', 'Fuel Consumption - Conventional');
h = plot(FC_Conv_Reg, xData, yData );
legend( h, 'Fuel Consumption vs. Capacity',...
'Fuel Consumption - Conventional', 'Location', 'NorthEast' );

% Label axes
xlabel('Capacity [TEU]')
ylabel('Fuel Consumption [t/h]')
grid on
grid minor
set(gca,'FontName','Utopia');
set(gca,'FontSize', 16);

%===========================================================
%Speed
```

```
%============================================================

SP = xlsread('seaweb.xls',3,'A2:B1033');

TEUclass = SP(:,1);
Speed = SP(:,2);

[xData, yData] = prepareCurveData( TEUclass, Speed );

% Set up fittype and options
ft = fittype( 'poly1' );
opts = fitoptions( 'Method', 'LinearLeastSquares' );
opts.Robust = 'LAR';

% Fit model to data.
[Speed_reg, gof] = fit(xData, yData, ft, opts);

% Plot fit with data.
figure( 'Name', 'Speed - Conventional');
h = plot(Speed_reg, xData, yData );
legend( h, 'Speed vs. Capacity', 'Speed - Conventional', ...
    'Location', 'NorthEast' );

% Label axes
xlabel('Capacity [TEU]')
ylabel('Speed [kn]')
grid on
grid minor
set(gca,'FontName','Utopia');
set(gca,'FontSize', 16);

%============================================================
% Autonomous Fleet
%============================================================
%Newbuilding price
%============================================================

%Considered decrease in newbuilding price of 7%
NB_Price_Aut = NB_Price * (1-0.07);

num = xlsread('nb.xlsx');
TEUclass = num(:,1);
```

```
[xData, yData] = prepareCurveData( TEUclass, NB_Price_Aut );

% Set up fittype and options
ft = fittype( 'poly1' );
opts = fitoptions( 'Method', 'LinearLeastSquares' );
opts.Robust = 'LAR';

% Fit model to data.
[NB_price_aut_reg, gof] = fit(xData, yData, ft, opts );

% Plot fit with data.
figure( 'Name', 'Newbudiling Price - Autonomous');
h = plot(NB_price_aut, xData, yData );
legend( h, 'Newbuilding price vs. Capacity', ...
'Newbudiling Price - Autonomous', 'Location', 'NorthEast' );

% Label axes
xlabel('Capacity [TEU]')
ylabel('Newbuilding price [USD]')
grid on
grid minor
set(gca,'FontName','Utopia');
set(gca,'FontSize', 16);

%==========================================================
%Fuel consumption
%==========================================================

%Considered decrease in newbuilding price of 6.3%
FC_Aut = FC_conv * (1-0.063);

FC = xlsread('seaweb.xls',2,'A1:B92');
TEUclass = FC(:,1);

[xData, yData] = prepareCurveData( TEUclass, FC_Aut );

% Set up fittype and options
ft = fittype( 'poly1' );
opts = fitoptions( 'Method', 'LinearLeastSquares' );
opts.Robust = 'LAR';

% Fit model to data.
[FC_aut_reg, gof] = fit(xData, yData, ft, opts );
```

```
%Plot fit with data.
figure( 'Name', 'Fuel Consumption - Conventional');
h = plot(FC_Conv_Reg, xData, yData );
legend( h, 'Fuel Consumption vs. Capacity',....
'Fuel Consumption - Conventional', 'Location', 'NorthEast' );

% Label axes
xlabel('Capacity [TEU]')
ylabel('Fuel Consumption [t/h]')
grid on
grid minor
set(gca,'FontName','Utopia');
set(gca,'FontSize', 16);

%===========================================================
%OPEX
%===========================================================

 OpEx_Aut = OpEx * (1-0.52);
 RegOPEX_Aut = fit(TEUClassAll', OpEx_Aut', 'poly1');
```

B.4 Multi-Epoch Analysis

```
% mea.m - Multi-Epoch Analysis
%= Written by Harald Ulvestad Salvesen, Spring 2018
%= Norwegian University of Science and Technology
clear all;clc;
%E0
bunkerPrice = 625; %[USD/ton]

%Load regression models
getReg();

%=============================================
% 2 Vessel Fleet
%=============================================
%Load results
load('2v300501000.mat');
Capacities2v = combsCap;
Distances2v = resultDist;
PortCall2v = resultPC;

fleetResults2v.Capacities = Capacities2v;
fleetResults2v.Distances = Distances2v;
fleetResults2v.PortCall = PortCall2v;

[nFleets nVessels] = size(fleetResults2v.Capacities);

%Conventional
%Calculate fleet cost
t = 1;
for f = 1:nFleets
    for v = 1:nVessels

        %Determine design speed for each vessel
        dSpeed = Speed_reg(fleetResults2v.Capacities(f,v)); ...
           %[kn = nm/h]
        %Determine sailing time
        timeAtSea = (fleetResults2v.Distances(f,v)/dSpeed); ...
           %[hours]
        %Calculate deisgn fuel cost
        fuelCost = ...
           FC_Conv_Reg(fleetResults2v.Capacities(f,v))......
```

```
        *timeAtSea*bunkerPrice; %[tons]
        %Calculate OPEX
        OPEX = RegOPEX(fleetResults2v.Capacities(f,v))*timeAtSea;

        Cost(v) = fuelCost + OPEX + fleetResults2v.PortCall(f,v);

        %Calculate CAPEX
        CAPEX(v) = NB_Price_Reg(fleetResults2v.Capacities(f,v));

    end
    OPEXVOYEX2v_conv(f) = sum(Cost)/10^6;
    CAPCOST2v_conv(f) = sum(CAPEX)/10^6;
    traceFleet(t,1) = OPEXVOYEX2v_conv(f) + CAPCOST2v_conv(f);
    traceFleet(t,2) = t;
    traceFleet(t,3) = OPEXVOYEX2v_conv(f);
    traceFleet(t,4) = CAPCOST2v_conv(f);
    t = t + 1;
    clear Cost; clear CAPEX;
end

hold on
sz=85;
scatter(OPEXVOYEX2v_conv,CAPCOST2v_conv,sz,'blue');


%Autonomous
%Calculate fleet cost

for f = 1:nFleets
    for v = 1:nVessels

        %Determine design speed for each vessel
        dSpeed = Speed_reg(fleetResults2v.Capacities(f,v)); ...
            %[kn = nm/h]
        %Determine sailing time
        timeAtSea = (fleetResults2v.Distances(f,v)/dSpeed); ...
            %[hours]
        %Calculate deisgn fuel cost
        fuelCost = FC_aut_reg(fleetResults2v.Capacities(f,v))*...
        timeAtSea*bunkerPrice; %[tons]
        %Calculate OPEX
        OPEX = ...
            RegOPEX_Aut(fleetResults2v.Capacities(f,v))*timeAtSea;
```

```
        Cost(v) = fuelCost + OPEX + fleetResults2v.PortCall(f,v);

        %Calculate CAPEX
        CAPEX(v) = ...
           NB_price_aut_reg(fleetResults2v.Capacities(f,v));

    end
    OPEXVOYEX2v_aut(f) = sum(Cost)/10^6;
    CAPCOST2v_aut(f) = sum(CAPEX)/10^6;
    traceFleet(t,1) = OPEXVOYEX2v_aut(f) + CAPCOST2v_aut(f);
    traceFleet(t,2) = t;
    traceFleet(t,3) = OPEXVOYEX2v_aut(f);
    traceFleet(t,4) = CAPCOST2v_aut(f);
    t = t + 1;
    clear Cost; clear CAPEX;
end

scatter(OPEXVOYEX2v_aut,CAPCOST2v_aut,sz,'blue','x')

clear combsCap; clear resultDist; clear resultPC;

%======================================
% 3 Vessel Fleet
%======================================

%Load results from fleets with 3 vessels
load('3v10050700.mat');
Capacities3v = combsCap;
Distances3v = resultDist;
PortCall3v = resultPC;

fleetResults3v.Capacities = Capacities3v;
fleetResults3v.Distances = Distances3v;
fleetResults3v.PortCall = PortCall3v;

[nFleets nVessels] = size(fleetResults3v.Capacities);

%Conventional Container Vessels
%Calculate fleet cost

for f = 1:nFleets
    for v = 1:nVessels
```

```
        %Determine design speed for each vessel
        dSpeed = Speed_reg(fleetResults3v.Capacities(f,v)); ...
           %[kn = nm/h]
        %Determine sailing time
        timeAtSea = (fleetResults3v.Distances(f,v)/dSpeed); ...
           %[hours]
        %Calculate deisgn fuel cost
        fuelCost = ...
           FC_Conv_Reg(fleetResults3v.Capacities(f,v))*....
        timeAtSea*bunkerPrice; %[tons]
        %Calculate OPEX
        OPEX = RegOPEX(fleetResults3v.Capacities(f,v))*timeAtSea;

        Cost(v) = fuelCost + OPEX + fleetResults3v.PortCall(f,v);

        %Calculate CAPEX
        CAPEX(v) = NB_Price_Reg(fleetResults3v.Capacities(f,v));

    end
    OPEXVOYEX3v_conv(f) = sum(Cost)/10^6;
    CAPCOST3v_conv(f) = sum(CAPEX)/10^6;
    traceFleet(t,1) = OPEXVOYEX3v_conv(f) + CAPCOST3v_conv(f);
    traceFleet(t,2) = t;
    traceFleet(t,3) = OPEXVOYEX3v_conv(f);
    traceFleet(t,4) = CAPCOST3v_conv(f);
    t = t + 1;
    clear Cost; clear CAPEX;
end

scatter(OPEXVOYEX3v_conv,CAPCOST3v_conv,sz,'red')

%Autonomous
%Calculate fleet cost

for f = 1:nFleets
    for v = 1:nVessels

        %Determine design speed for each vessel
        dSpeed = Speed_reg(fleetResults3v.Capacities(f,v)); ...
           %[kn = nm/h]
        %Determine sailing time
```

```
        timeAtSea = (fleetResults3v.Distances(f,v)/dSpeed); ...
           %[hours]
        %Calculate deisgn fuel cost
        fuelCost = ...
           FC_aut_reg(fleetResults3v.Capacities(f,v))*.....
        timeAtSea*bunkerPrice; %[tons]
        %Calculate OPEX
        OPEX = ...
           RegOPEX_Aut(fleetResults3v.Capacities(f,v))*timeAtSea;

        Cost(v) = fuelCost + OPEX + fleetResults3v.PortCall(f,v);

        %Calculate CAPEX
        CAPEX(v) = ...
           NB_price_aut_reg(fleetResults3v.Capacities(f,v));

    end
    OPEXVOYEX3v_aut(f) = sum(Cost)/10^6;
    CAPCOST3v_aut(f) = sum(CAPEX)/10^6;
    traceFleet(t,1) = OPEXVOYEX3v_aut(f) + CAPCOST3v_aut(f);
    traceFleet(t,2) = t;
    traceFleet(t,3) = OPEXVOYEX3v_aut(f);
    traceFleet(t,4) = CAPCOST3v_aut(f);
    t = t + 1;
    clear Cost; clear CAPEX;
end

scatter(OPEXVOYEX3v_aut,CAPCOST3v_aut,sz,'red','x')

clear combsCap; clear resultDist; clear resultPC;

%==============================

%Load results from fleets with 4 vessels
load('4v10050550.mat');
Capacities4v = combsCap;
Distances4v = resultDist;
PortCall4v = resultPC;

fleetResults4v.Capacities = Capacities4v;
fleetResults4v.Distances = Distances4v;
fleetResults4v.PortCall = PortCall4v;
```

```
[nFleets nVessels] = size(fleetResults4v.Capacities);

%Conventional Container Vessels
%Calculate fleet cost

for f = 1:nFleets
    for v = 1:nVessels

        %Determine design speed for each vessel
        dSpeed = Speed_reg(fleetResults4v.Capacities(f,v)); ...
           %[kn = nm/h]
        %Determine sailing time
        timeAtSea = (fleetResults4v.Distances(f,v)/dSpeed); ...
           %[hours]
        %Calculate deisgn fuel cost
        fuelCost = ...
           FC_Conv_Reg(fleetResults4v.Capacities(f,v))*....
        timeAtSea*bunkerPrice; %[tons]
        %Calculate OPEX
        OPEX = RegOPEX(fleetResults4v.Capacities(f,v))*timeAtSea;

        Cost(v) = fuelCost + OPEX + fleetResults4v.PortCall(f,v);

        %Calculate CAPEX
        CAPEX(v) = NB_Price_Reg(fleetResults4v.Capacities(f,v));

    end
    OPEXVOYEX4v_conv(f) = sum(Cost)/10^6;
    CAPCOST4v_conv(f) = sum(CAPEX)/10^6;
    traceFleet(t,1) = OPEXVOYEX4v_conv(f) + CAPCOST4v_conv(f);
    traceFleet(t,2) = t;
    traceFleet(t,3) = OPEXVOYEX4v_conv(f);
    traceFleet(t,4) = CAPCOST4v_conv(f);
    t = t + 1;
    clear Cost; clear CAPEX;
end


scatter(OPEXVOYEX4v_conv,CAPCOST4v_conv,sz,'green')

%Autonomous
%Calculate fleet cost
```

```matlab
for f = 1:nFleets
    for v = 1:nVessels

        %Determine design speed for each vessel
        dSpeed = Speed_reg(fleetResults4v.Capacities(f,v)); ...
           %[kn = nm/h]
        %Determine sailing time
        timeAtSea = (fleetResults4v.Distances(f,v)/dSpeed); ...
           %[hours]
        %Calculate deisgn fuel cost
        fuelCost = ...
           FC_aut_reg(fleetResults4v.Capacities(f,v))*....
        timeAtSea*bunkerPrice; %[tons]
        %Calculate OPEX
        OPEX = ...
           RegOPEX_Aut(fleetResults4v.Capacities(f,v))*timeAtSea;

        Cost(v) = fuelCost + OPEX + fleetResults4v.PortCall(f,v);

        %Calculate CAPEX
        CAPEX(v) = ...
           NB_price_aut_reg(fleetResults4v.Capacities(f,v));

    end
    OPEXVOYEX4v_aut(f) = sum(Cost)/10^6;
    CAPCOST4v_aut(f) = sum(CAPEX)/10^6;
    traceFleet(t,1) = OPEXVOYEX4v_aut(f) + CAPCOST4v_aut(f);
    traceFleet(t,2) = t;
    traceFleet(t,3) = OPEXVOYEX4v_aut(f);
    traceFleet(t,4) = CAPCOST4v_aut(f);
    t = t + 1;
    clear Cost; clear CAPEX;
end

scatter(OPEXVOYEX4v_aut,CAPCOST4v_aut,sz,'green','x')
grid on
grid minor
xlabel('OPEX & VOYEX [millUSD]')
ylabel('CAPEX [millUSD]')
ylim([15 50]);
set(gca,'FontName','Utopia');
set(gca,'FontSize', 24);
```

```
LH(1) = plot(nan, nan, 'ob');
L{1} = 'Conv. - 2 vessels';
LH(2) = plot(nan, nan, 'or');
L{2} = 'Conv. - 3 vessels';
LH(3) = plot(nan, nan, 'og');
L{3} = 'Conv. - 4 vessels';
LH(4) = plot(nan, nan, 'xb');
L{4} = 'Auto.  - 2 vessels';
LH(5) = plot(nan, nan, 'xr');
L{5} = 'Auto.  - 3 vessels';
LH(6) = plot(nan, nan, 'xg');
L{6} = 'Auto.  - 4 vessels';

legend(LH,L,'Location','northeast')

hold off
hold on

%Trace designs with min. costs

sortFleet = sort(traceFleet(:,1));
top = sortFleet(1:25);
for j = 1:numel(top)
paretoFleets(j) = find(traceFleet(:,1) == top(j));
end
paretoFleets = sort(paretoFleets)'

xlswrite('pareto.xls',paretoFleets,'I1:I25')

sz=160;
scatter(traceFleet(paretoFleets,3),....
traceFleet(paretoFleets,4),sz,'m')
hold off
```

B.5 Baltic Trade Network

```
%= Baltic Trade Network data
%= Written by Harald Ulvestad Salvesen, Spring 2018
%= Norwegian University of Science and Technology
edgeList = {[2 3 4 5 6 7 8] [1 3 4 5 6 7 8] [1 2 4 5 6 7 8]...
    [1 2 3 5 6 7 8] [1 2 3 4 6 7 8] [1 2 3 4 5 7 8] [1 2 3 4 ...
       5 6 8] ...
    [1 2 3 4 5 6 7];[1 2 3 4 5 6 7] [1 8 9 10 11 12 13]...
    [2 8 14 15 16 17 18] [3 9 14 19 20 21 22] [4 10 15 19 23 ...
       24 25]...
    [5 11 16 20 23 26 27] [6 12 17 21 24 26 28] [7 13 18 22 ...
       25 27 28]};

FullMap = graph({'DEBRV','DEBRV','DEBRV',...
'DEBRV','DEBRV','DEBRV','DEBRV'...
    },{'RUKGD','FIKTK','FIRAU','NOAES','NOBGO','NOKRS','NOSVG'});

FullMap.Edges.Var2(1,1) = 832;
FullMap.Edges.Var2(2) = 1075;
FullMap.Edges.Var2(3) = 1060;
FullMap.Edges.Var2(4) = 545;
FullMap.Edges.Var2(5) = 447;
FullMap.Edges.Var2(6) = 292;
FullMap.Edges.Var2(7) = 366;
FullMap.Edges.Properties.VariableNames{2} = 'Distance';

FullMap = addedge(FullMap, ...
   {'RUKGD','RUKGD','RUKGD','RUKGD','RUKGD','RUKGD'},{'FIKTK',...
    'FIRAU','NOAES','NOBGO','NOKRS','NOSVG'});

FullMap.Edges.Distance(8) = 475;
FullMap.Edges.Distance(9) = 678;
FullMap.Edges.Distance(10) = 962;
FullMap.Edges.Distance(11) = 817;
FullMap.Edges.Distance(12) = 604;
FullMap.Edges.Var3(12,1) = 733;
FullMap.Edges.Var3(12) = 0;
FullMap.Edges.Distance(13) = 348;

FullMap = addedge(FullMap, ...
   {'FIKTK','FIKTK','FIKTK','FIKTK','FIKTK',},{'FIRAU','NOAES'...
```

```
    ,'NOBGO','NOKRS','NOSVG'});

FullMap.Edges.Distance(14) = 541;
FullMap.Edges.Distance(15) = 1178;
FullMap.Edges.Distance(16) = 1062;
FullMap.Edges.Distance(17) = 847;
FullMap.Edges.Distance(18) = 976;

 FullMap = addedge(FullMap, ...
    {'FIRAU','FIRAU','FIRAU','FIRAU',},{'NOAES','NOBGO',.......
     'NOKRS','NOSVG'});

FullMap.Edges.Distance(19) = 1190;
FullMap.Edges.Distance(20) = 1045;
FullMap.Edges.Distance(21) = 832;
FullMap.Edges.Distance(22) = 961;

FullMap = addedge(FullMap, ...
   {'NOAES','NOAES','NOAES',},{'NOBGO','NOKRS',.....
    'NOSVG'});

FullMap.Edges.Distance(23) = 167;
FullMap.Edges.Distance(24) = 393;
FullMap.Edges.Distance(25) = 278;

FullMap = addedge(FullMap, {'NOBGO','NOBGO'},{'NOKRS','NOSVG'});

FullMap.Edges.Distance(26) = 226;
FullMap.Edges.Distance(27) = 111;

FullMap = addedge(FullMap, {'NOKRS'},{'NOSVG'});

FullMap.Edges.Distance(28) = 142;

for i = 1:8
    FullMap.Nodes.Var2(i) = i;
end

for i = 1:28
FullMap.Edges.Var3(i) = i;
end

FullMap.Nodes.Var3(2,1) = 536;
```

```
FullMap.Nodes.Var3(3) = 374;
FullMap.Nodes.Var3(4) = 36;
FullMap.Nodes.Var3(5) = 20;
FullMap.Nodes.Var3(6) = 34;
FullMap.Nodes.Var3(7) = 12;
FullMap.Nodes.Var3(8) = 130;

FullMap.Edges.Properties.VariableNames{3} = 'EdgeID';
FullMap.Nodes.Properties.VariableNames{2} = 'NodeID';
FullMap.Nodes.Properties.VariableNames{3} = 'Demand';


FullMap.Edges.Var4(1,1) = 1;
FullMap.Edges.Var4(2) = 1;
FullMap.Edges.Var4(3) = 1;
FullMap.Edges.Var4(4) = 1;
FullMap.Edges.Var4(5) = 1;
FullMap.Edges.Var4(6) = 1;
FullMap.Edges.Var4(7) = 1;
FullMap.Edges.Var5(1,1) = 2;
FullMap.Edges.Var5(2) = 3;
FullMap.Edges.Var5(3) = 4;
FullMap.Edges.Var5(4) = 5;
FullMap.Edges.Var5(5) = 6;
FullMap.Edges.Var5(6) = 7;
FullMap.Edges.Var5(7) = 8;
FullMap.Edges.Var4(8) = 2;
FullMap.Edges.Var4(9) = 2;
FullMap.Edges.Var4(10) = 2;
FullMap.Edges.Var4(11) = 2;
FullMap.Edges.Var4(12) = 2;
FullMap.Edges.Var4(13) = 2;
FullMap.Edges.Var5(8) = 3;
FullMap.Edges.Var5(9) = 4;
FullMap.Edges.Var5(10) = 5;
FullMap.Edges.Var5(11) = 6;
FullMap.Edges.Var5(12) = 7;
FullMap.Edges.Var5(13) = 8;
FullMap.Edges.Var4(14) = 3;
FullMap.Edges.Var4(15) = 3;
FullMap.Edges.Var4(16) = 3;
FullMap.Edges.Var4(17) = 3;
FullMap.Edges.Var4(18) = 3;
```

```
FullMap.Edges.Var5(14) = 4;
FullMap.Edges.Var5(15) = 5;
FullMap.Edges.Var5(16) = 6;
FullMap.Edges.Var5(17) = 7;
FullMap.Edges.Var5(18) = 8;
FullMap.Edges.Var4(19) = 4;
FullMap.Edges.Var4(20) = 4;
FullMap.Edges.Var4(21) = 4;
FullMap.Edges.Var4(22) = 4;
FullMap.Edges.Var5(19) = 5;
FullMap.Edges.Var5(20) = 6;
FullMap.Edges.Var5(21) = 7;
FullMap.Edges.Var5(22) = 8;
FullMap.Edges.Var4(23) = 5;
FullMap.Edges.Var4(24) = 5;
FullMap.Edges.Var4(25) = 5;
FullMap.Edges.Var5(23) = 6;
FullMap.Edges.Var5(24) = 7;
FullMap.Edges.Var5(25) = 8;
FullMap.Edges.Var4(26) = 6;
FullMap.Edges.Var4(27) = 6;
FullMap.Edges.Var5(26) = 7;
FullMap.Edges.Var5(27) = 8;
FullMap.Edges.Var4(28) = 7;
FullMap.Edges.Var5(28) = 8;
FullMap.Edges.Properties.VariableNames{4} = 'Start';
FullMap.Edges.Properties.VariableNames{5} = 'End';

%Add port call cost
FullMap.Nodes.Var4(1,1) = 0;
FullMap.Nodes.Var4(2) = 1062;
FullMap.Nodes.Var4(3) = 1182;
FullMap.Nodes.Var4(4) = 18552;
FullMap.Nodes.Var4(5) = 24098;
FullMap.Nodes.Var4(6) = 17435;
FullMap.Nodes.Var4(7) = 24076;
FullMap.Nodes.Var4(8) = 1227;
FullMap.Nodes.Properties.VariableNames{4} = 'PortCall';
plot(FullMap);
```

B.6 Optimal Fleet Trace

```
%= Optimal Fleet Trace from the Multi-Epoch Analysis
%= Written by Harald Ulvestad Salvesen, Spring 2018
%= Norwegian University of Science and Technology
paretodata = xlsread('pareto.xls');

paretodata = paretodata(:);
paretodata = sort(paretodata);

[a,b] = hist(paretodata,unique(paretodata));

fleetIDs = unique(paretodata);

h = histogram(paretodata, 1:95);
h.EdgeColor = 'b';
grid on
xlabel('Fleet ID')
xlim([1 762])
ylim([0 28])
xticks([0:50:762])
yticks([0:2:28])
ylabel('Frequency of Occurence')
set(gca,'FontName','Utopia');
set(gca,'FontSize', 24);
view([90 -90])

hold on
g = histogram(paretodata, 96:190)
g.EdgeColor = 'r';
hold off

hold on
g = histogram(paretodata, 191:392)
g.EdgeColor = 'b';
hold off

hold on
g = histogram(paretodata, 393:594)
g.EdgeColor = 'r';
hold off
```

```
hold on
g = histogram(paretodata, 595:678)
g.EdgeColor = 'b';
hold off


hold on
g = histogram(paretodata, 679:762)
g.EdgeColor = 'b';
hold off
```

Appendix C

Xpress-IVE Code

C.1 TSP Model

The Xpress code for the TSP model starts on the next page.

```
! Written by Harald Ulvestad Salvesen, Spring 2017
! Norwegian University of Science and Technology

model TSPmod2
uses "mmxprs"; !gain access to the Xpress-Optimizer solver
options explterm   !Line breaks is not an expression separator. Each expression
must end with a ;
options noimplicit !Everything except indices must be declared BEFORE it is used

!Load parameters from file
parameters
    DataFile = "TSPData.txt";
end-parameters

declarations
   nNodes: integer;
   nEdge: integer;
   nEdgeSet: integer;
   nDist: integer;
   nCombs: integer;
end-declarations

initializations from DataFile
    nNodes;
    nEdge;
    nEdgeSet;
    nDist;
    nCombs;
end-initializations

declarations
    W: set of integer;
    Combs: set of integer;
end-declarations

W := 2..nNodes;
Combs := 1..nCombs;

finalize(W);
finalize(Combs);

declarations
    EdgeSet: set of integer;
    E: array(W, Combs) of set of integer;
end-declarations

initializations from DataFile
    EdgeSet;
    E;
end-initializations

forall(ii in W, jj in Combs) finalize(E(ii,jj));

!Declaring sets:
declarations
    Nodes: set of integer;
    Edges: set of integer;
    DistIndex: set of integer;
end-declarations

Nodes := 1..nNodes;
Edges := 1..nEdge;
DistIndex := 1..nDist;

finalize(Nodes);
finalize(Edges);
finalize(EdgeSet);
finalize(DistIndex);
```

```
declarations
    Distance: array(DistIndex) of integer;
    Delta: array(Nodes, Edges) of integer;
end-declarations

initializations from DataFile
    Distance;
    Delta;
end-initializations

!Declaring indexed decision variable as dynamic array
declarations
    flow: dynamic array(EdgeSet) of mpvar;
end-declarations

forall (ee in EdgeSet) do
  create(flow(ee));
  flow(ee) is_binary;
end-do

declarations
    TotalDist: linctr;
    C1: array(Nodes) of linctr;
    C2: array(W,Combs) of linctr;
end-declarations

!Obj. function
!=========================
TotalDist :=
  sum(ee in EdgeSet) Distance(ee)*flow(ee);

!C1
!=========================
forall (ii in Nodes) do
  C1(ii) := sum(jj in Edges) flow(Delta(ii,jj)) = 2;
end-do

!C2 - Subtour constraint
!=========================
forall (i in W) do
    forall (j in Combs)
        C2(i,j) := sum(k in E(i,j)) flow(k) <= i - 1;
end-do

minimize(TotalDist);

initializations to "matlab.mws:"
    evaluation of getobjval as "objval";
    evaluation of array(ii in EdgeSet) flow(ii) .sol as "flow";
end-initializations

end-model
```

C.2 VRP - IP Model

The Xpress code for the IP model starts on the next page.

```
! Written by Harald Ulvestad Salvesen, Spring 2017
! Norwegian University of Science and Technology

model VRPMod
uses "mmxprs"; !gain access to the Xpress-Optimizer solver
options explterm   !Line breaks is not an expression separator. Each expression
must end with a ;
options noimplicit !Everything except indices must be declared BEFORE it is used


parameters
    DataFile = 'VRPData.txt';
end-parameters

declarations
    nNodes: integer;
    nVessels: integer;
    nRoutes: integer;
    maxTime: integer;
end-declarations

initializations from DataFile
    nNodes;
    nVessels;
    nRoutes;
    maxTime;
end-initializations

declarations
    Nodes: set of integer;
    Vessels: range;
    Routes: set of integer;
end-declarations

Nodes := 1..nNodes;
Vessels := 1..nVessels;
Routes := 1..nRoutes;

finalize(Nodes);
finalize(Vessels);
finalize(Routes);

declarations
    R: dynamic array(Vessels) of set of integer;
    C: dynamic array(Vessels, Routes) of real;
    T: dynamic array(Vessels, Routes) of integer;
    A: dynamic array(Vessels, Routes, Nodes) of integer;
end-declarations

initializations from DataFile
    C; A; R; T;
end-initializations

forall(vv in Vessels) finalize(R(vv));

declarations
    x: dynamic array(Vessels, Routes) of mpvar;
end-declarations

forall(vv in Vessels, rr in R(vv) | exists(T(vv,rr)) and T(vv,rr)<>0 and
exists(C(vv,rr)) and C(vv,rr)<>0 and
                (or(nn in Nodes | exists(A(vv,rr,nn))) true) ) do
    create(x(vv,rr));
    x(vv,rr) is_binary;
end-do

declarations
    Obj: linctr;
```

```
    C1: array(Nodes) of linctr;
    C2: array(Vessels) of linctr;
    C3: array(Vessels) of linctr;
end-declarations

Obj := sum(vv in Vessels, rr in R(vv) | exists(x(vv,rr))) (C(vv,rr)*x(vv,rr));

!Constraint 1
forall (nn in Nodes) do
    C1(nn):= sum(vv in Vessels, rr in R(vv) | exists(x(vv,rr)))
A(vv,rr,nn)*x(vv,rr) >= 1;
end-do

!Constraint 2
forall (vv in Vessels) do
    C2(vv):= sum(rr in R(vv)) x(vv,rr) >= 1;
end-do

!Constraint 3
forall (vv in Vessels) do
    C3(vv) := sum(rr in R(vv) | exists(x(vv,rr))) T(vv,rr)*x(vv,rr) <= maxTime;
end-do

!Minimize objective function
!=====================================================================
minimize(Obj);

!Parse optimal distance to MATLAB
!===================================================================
initializations to "matlab.mws:"
    evaluation of getobjval as "vrpsol";
    evaluation of array(vv in Vessels, rr in R(vv)) x(vv,rr) .sol as "vrpflow";
end-initializations

end-model
```