Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Marine Technology

Roald Hartvigsen

Inventory Managemement, Scheduling and Routing in Fish Feed Distribution

Master's thesis in Marine Technology
Supervisor: Bjørn Egil Asbjørnslett, IMT
June 2019

NTNU
Norwegian University of
Science and Technology

Roald Hartvigsen

Inventory Managemement, Scheduling and Routing in Fish Feed Distribution

Master’s thesis in Marine Technology
Supervisor: Bjørn Egil Asbjørnslett, IMT
June 2019

Norwegian University of Science and Technology
Faculty of Engineering
Department of Marine Technology

NTNU

Master`s Agreement

Faculty	IV - Fakultet for ingeniørvitenskap
Institute	Institutt for marin teknikk
Programme code	MTMART
Course code	194_TMR4930_1

Personal information

Family name, first name	Hartvigsen, Roald
Date of birth	04.10.1994
Email address	roalh@stud.ntnu.no

The Master`s thesis

Starting date	15.01.2019
Submission deadline	11.06.2019
Thesis working title	Inventory Management, Scheduling and Routing in Fish Feed Distribution
Thematic description	The purpose of this master thesis is to develop models used for optimization and solution methods for said problems. The optimization models will represent Mowi ASA's fish feed transportation, from their location at Valneset, Bjugn and to their fish farming locations along the Norwegian coast. The optimization model shall take ship routing and scheduling into account for a given fleet of ships. Realistic constraints and inventory restrictions should also be considered when modeling the problem. Solution methods shall be implemented in order to test their performance relative to each other for given model complexities, given by the time horizon and the number of nodes. Main contents: - Problem description - Literature review for similar optimization problems - Model considerations, including considerations for objective and constraints - Mathematical formulation of problem - Presentation of optimization algorithms and their implementation - Computational performance of different formulations and optimization algorithms - Discussion of findings and further work

Supervision and co-authors

Supervisor	**Bjørn Egil Asbjørnslett**
Any co-supervisors	
Any co-authors	

Topics to be included in the Master`s Degree (if applicable)

Guidelines – Rights and Obligations

Purpose

Agreement on supervision of the Master's thesis is a cooperation agreement between the student, supervisor and the department that governs the relationship of supervision, scope, nature and responsibilities.

The master's program and the work of the master's thesis are regulated by the Act relating to universities and university colleges, NTNU's study regulations and current curriculum for the master's program.

Supervision

The student is responsible for

- Agre upon supervision within the framework of the agreement
- Set up a plan of progress for the work in cooperation with the supervisor, including the plan for when the guidance should take place
- Keep track of the number of hours spent with the supervisor
- Provide the supervisor with the necessary written material in a timely manner before the guidance
- Keep the institute and supervisor informed of any delays

The supervisor is responsible for

- Explain expectations of the guidance and how the guidance should take place
- Ensure that any necessary approvals are requested (REC, ethics, privacy)
- Provide advice on the formulation and demarcation of the topic and issue so that the work is feasible within the standard or agreed upon study time
- Discuss and evaluate hypotheses and methods
- Advice on professional literature, source material / data base / documentation and potential resource requirements
- Discuss the presentation (disposition, linguistic form, etc.)
- Discuss the results and the interpretation of them
- Stay informed about the progression of the student's work according to the agreed time and work plan, and follow up the student as needed
- Together with the student, keep an overview of the number of hours spent

The institute is responsible for

- Make sure that the agreement is entered into

- Find and appoint supervisor(-s)
- Enter into an agreement with another department / faculty / institution if there is a designated external supervisor
- In cooperation with the supervisor, keep an overview of the student's progress, an overview of the number of hours spent, and follow up if the student is delayed by appointment
- Appoint a new supervisor and arrange for a new agreement if
    - supervisor will be absent due to research term, illness, travel, etc., and if the student wishes
    - student or supervisor requests to terminate the agreement because one of the parties does not follow it
    - other circumstances make the parties find it appropriate with a new supervisor
- Notify the student when the guidance relationship expires.
- Inform supervisors about the responsibility for safeguarding ethical issues, privacy and guidance ethics
- Should the cooperation between student and supervisor become problematic for one of the parties, a student or supervisor may ask to be freed from the Master`s agreement. In such case, the institute must appoint a new supervisor

This Master`s agreement must be signed when the guidelines have been reviewed.

Signatures

Institute	**Supervisor**	**Student**
IMT, 03.05.19		
place and date	place and date	place and
date [illegible]	Trondheim 25.04.19	Trondheim 25.04.19
	[illegible]	[illegible]

Preface

This masters thesis is the finalization of a Masters of Science degree at the Norwegian University of Science and Technology, Trondheim. The degree specialization is Marine Systems Design and Logistics at the Department of Marine Technology. The thesis is written during the spring of 2019.

I would like to thank my academic supervisor Professor Bjørn Egil Asbjørnslett. Your have given excellent guidance and feedback, that has helped structure the thesis. Your insights regarding the complexity of the distribution system has also been helpful. Furthermore, the thesis would not have been possible without the substantial work on clustering of fish farms done by Marius Gyberg Haugland and Sondre Thygesen.

Trondheim, June 11th, 2019

Roald Hartvigsen

Roald Hartvigsen

Summary

This master's thesis addresses routing of ships, used in order to serve an industrial network of nodes covering aquaculture locations along the Norwegian coastline. Aquaculture locations are served with feed from a central factory with self-operated vessels or with externally produced feed by external vessel. The aim of the master's thesis is making a routing model that minimizes operating costs, while ensuring feed deliveries that meet the feed requirements at the fish farms at any given time. Fish feed is one of the main cost drivers in the salmon farming industry. Failure to provide sufficient amount of feed at the right time can impact the growth of the salmon and therefore also reduce revenues. Reliable deliveries are therefore crucial in order to keep unit costs low in the salmon farming industry. The literature study found that substantial work have been done on Vehicle Routing Problem with maritime applications. This includes route generation methods applied in the aquaculture industry for feed deliveries. There are also literature on Inventory Routing Problem with maritime applications. Evolutionary algorithms, cross-entropy methods and decomposition methods are examples of solution methods used to solve IRPs in existing literature.

An Inventory Routing Model is formulated in order to include both inventory management and routing in the planning process. Both arc-load and arc-flow models of the problem are implemented, and further formulation improvements are considered. This includes sub-tour elimination constraints and clique inequalities. In addition a Dantzig-Wolfe decomposition is formulated. Inventory Routing Problems are modelled with multiple nodes at each harbour in order to incorporate the possibility of multiple deliveries at a harbour. Thus, the computational complexity grows rapidly. Model formulations and formulation improvements are therefore essential in order to keep computation times down, and to provide better solutions and bounds. All model formulations are applied on test cases where fish farms are clustered into a varying number of nodes. Model formulations are also applied in order to test consumption scenarios, compatibility scenarios and robustness.

The arc-flow model consistently generated superior solutions and bounds for all test cases, helped by the removal of Big M methods. Both the split and aggregated sub-tour elimination constraints, as well as the clique inequalities, gave significant improvements in solutions and bounds for the arc-load formulation, but had limited effect when coupled with the arc-flow formulation. By limiting compatibility between ships and fish farms, integrality gaps were significantly reduced. However, the solutions found were inferior as arcs of lower cost were neglected. A major part of the cost increases associated with limited compatibility could be avoided by small changes, as allowing large vessels to serve high demand clusters. Consumption scenarios were also tested. As expected, increased consumption lead to increased costs, as external deliveries were needed in order to serve all fish farms. Interesting topic for further research that are not included in this thesis, includes additions of penalty costs for late deliveries and modelling of stochastic behaviour.

This page is intentionally left blank.

Sammendrag

Denne masteroppgaven tar for seg ruting av skip, brukt til å betjene et industrielt nettverk av noder som dekker lokasjoner for lakseoppdrett langs norskekysten. Oppdrettslokasjonene betjenes med fôr som produseres ved en sentral fabrikk og fraktes med selveide skip. Målet med oppgaven er å formulere en routingmodell som minimerer operasjonskostnader, mens fôrleveransene er tilstrekkelig for å møte fôrforbruket ved oppdrettslokasjonene til enhver tid. Fiskefôr er en av kostnadsdriverne i oppdrettssektoren og mangel på fôrleveranser kan føre til mindre vekst blant laks, som igjen fører til tapt omsetning. Sikre og forutsigbare leveranser er derfor essensielt for å holde enhetskostnadene nede. Litteraturstudiet viste at det allerede er gjort mye arbeid med anvendelser av Vehicle Routing Problems og Inventory Routing Problems i maritime næringer. Dekomponering og cross-entropy algoritmer er noen av løsningsmetodene som er brukt for Inventory Routing Problemer i litteraturen.

En inventory routing model ble formulert for å inkludere både rutingen og administrasjon av fôrnivåer samtidig. Både en arc-load og en arc-flow modell ble implementer i Gurobi gjennom Matlab, og modellforbedringer ble vurdert. Modellforbedringene inkluderer sub-tour elimination constraints og clique inequalities. I tillegg er Dantzig-Wolfe dekomponering modellert. Invetory Routing problemet er modellert med flere noder for hver oppdrettslokasjon, og problemet blir derfor fort meget komplekst når et økende antall oppdrettslokasjoner inkluderes i problemet. Modellforbedringer er derfor essensielle for å holde tiden det tar å løse problemet nede, og for å generere gode løsninger og nedre grenser. Alle modellforbedringer er testet på eksempelsett, hvor oppdrettslokasjoner er gruppert sammen i grupper med forskjellige antall oppdrettslokasjoner i hver gruppe.

Arc-flow modellen genererte de beste løsningene uavhengig av hvilke modellforbedringer som ble testet. Eliminasjonen av constraints basert på Big M metoden var trolig avgjørende for at arc-flow modellen utkonkurrerte arc-load modellen. Sub-tour elimination constraints og clique inequalities gav betydelig forbedringer i løsningene for arc-load modellen, men hadde mindre effekt når de ble brukt med arc-flow modellen. Med å begrense hvilke skip som kunne besøke hvilke havner ble tidene det tok å løse problemene redusert betydelig. Dette kom på bekostning av økte kostnader, ettersom mindre kostbare rutealternativer ble neglisjert. Mye av de økte kostnadene kunne reverseres ved å tillate større skip å besøke oppdrettslokausjoner med høyt konsum. Testing av modellene ved endret fôrkonsum ble også utført, og viste som forventet at kostnadene økte ved økt konsum. Interessante emner for videre undersøkelse inkuderer straffekostnader ved forsikret levering og modellering av stokastiske prosesser.

This page is intentionally left blank.

Contents

List of Figures

List of Tables

1 Introduction

While humans have been consuming fish for more than 40 000 years (Hu et al., 2009), the techniques used to harvest the fish have been changing throughout history. The river Nile provided substantial amounts of food, and the Egyptians invented various fishing techniques (Kumar, 2001). Reed boats, woven nets, weir baskets, hooks and lines were used to capture the fish. The history of aquacultural traditions also stretches far back in time. Evidence suggests that indigenous people in Australia may have started farming eels about 8 000 years ago by developing floodplain into channels and dams (Salleh, 2003). Different techniques have been developed for farming of different species. According to FAO, farming of Atlantic Salmon is the fish farming industry that generates the most revenues (Fisheries, FAO, 2011). In 2016 the total revenues for all salmon farming companies and sub-segments in Norway totaled NOK 212.7 billions, while the total export value came in at NOK 65.2 billions (EY, 2017). Significant revenues have allowed the industry to develop fast in terms of technology, and the services used by the industry now include service vessels, live fish carriers, fish pellets carriers, vaccines and medicines among other thing. While these services often help increase revenues and often may be seen as an absolute necessity in order to farm salmon, inefficient use of services will decrease profits.

The Norwegian fish feed industry totaled NOK 24.5 billions in revenues in 2016, with a EBITDA margin of about 11% (EY, 2017). Production and transportation of feed are some of the key cost drivers of the industry. As the number of fish farming locations have increased, intuition is no longer sufficient when planing routing of the ships used when distributing feed. Thus optimization techniques implemented in the form of computer algorithms is necessary in order to evaluate and analyze routing problems in an efficient manner.

Maritime routing problems have been present for as long as ships have existed, and in recent years mathematical formulations have been used in order to solve these problem. Important work includes Fagerholt (1999) that solves a multi-trip vehicle routing problem VRPMT using a route generation algorithm and Christiansen (1999) which is formulating a maritime inventory routing problem. For transportation of fish feed, Haugland and Thygesen (2017) provides insight to the feed production and distribution in the aquaculture industry. Haugland and Thygesen formulates a VRP that is solved using the route generation algorithm laid out by Fagerholt.

The main objective of this paper is to formulate a mathematical model that can be solved in order to efficiently route ships in order to minimize operating costs and serve fish farming location, such that all important requirements of the fish farms are met. Requirements include that fish farms receive the correct type of feed before the storage runs out. The deliveries of feed at the fish farms can not exceed unfilled storage capacities. Only operating costs for the current fleet will be considered.

2 Problem formulation

For transportation problems it is necessary to be able to find efficient and robust solutions, that meets all the demands of the system. The aim of this paper is modelling and finding efficient optimization algorithms for routing of fish pellets carriers. The problem can be described as a variants of either the Vehicle Routing Problem or the Inventory Ship Routing Problem, were one at all times need to ensure that sufficient amounts of fish feed is available at the fish farming locations. Multiple model formulations will be tested in order to find the one producing the best solutions and bound. Formulation improvements will also be tested in order to see in solutions and bonds can be improved further. Ways of reducing the problem size and computational complexity will be explored, and tests of the models under different consumption scenarios and robustness requirements will be tested. In section 2.1 a description of the underlying feed distribution problem will be presented. Section 2.2 will present the objective of the paper, while section 2.3 will give details on the scope and limitations for the paper.

2.1 Case description

The fish farming company Mowi ASA produce fish feed from one central factory in Norway, that they distribute to approximately 150 different fish farming locations along the Norwegian coastline. The fish pellets factory is located at Valneset, Bjugn in the Trøndelag area, and have a nameplate production capacity of 400 000 tonnes a year. As seasonal demand is lower in the winter moths, the utilization is not at a maximum level at all times. Estimates for actual feed production are therefore in the range of 300 000 to 350 000 tonnes a year. Excess demand can be supplied by the external fish feed producers, as Biomar and Skretting. However, when order feed from external suppliers, the orders usually have to be placed weeks in advance. Thus, the current situation indicates that the planning of the feed distribution have to happen weeks before the actual deliveries.

Ships are used in order to transport the feed form the factory, and out to the fish farming locations. Mowi have a heterogeneous fleet available consisting of 4 vessels, and can also hire external vessel from other ship owners if needed. The fleet consists of two LNG powered vessels with a cargo capacity of 3 000 tonnes each, and two HFO powered vessels with a cargo capacity of 1 500 tonnes.

The process of deciding the routing of the ships today are time consuming and considered hard to model. It is often done by a combination of excel sheets and intuition, were vessels are allocated to different fish farming location based on the most recent needs. There are also restrictions limiting which vessel can serve which fish farming location. These restrictions are based on vessel speed, geographical location of the fish farms, offloading capabilities of vessels and their compatibility with the different fish farms. Different feed for different stages of production also makes the problem more complicated. The feed needed for smolt and smaller fish is different from the one used for larger fish. Also feed including antibiotics or other medicines may be used for fish in need of treatment. Singular unforeseen events can also impact consumption of feed, such as mass-escape of fish and illness. As including all these parameters in a single model would make it too computationally complex to be solved within a reasonable time frame, simplifications to the models have to be made. Feed distribution plans have to include the routing of the vessels. In other words, which vessels serve which fish farms, and in which order they do that. It also have to include the quantities of each feed type loaded at the feed factory and the quantities to be discharged at each fish farm. In addition, it has to be ensured that fish farms never run out of inventory. This can either be done by having excess inventory at the fish farms that ensures sufficient amounts of feed regardless of when arrivals happen, or by adding suitable delivery times to the feed distribution plans.

2.2 Problem objective

The objective of the paper is to find the optimal routing of ships used for fish feed distribution for Mowi ASA. The optimal routing, is defined as a routing that minimizes operating costs, while at the same time ensures that fish farming locations are able to run operations in an effective matter. The operating cost includes only fuel and external deliveries, as crew, port fees and other running costs are considered fixed. By running operations in an effective manner fish farms should always have sufficient amounts of the correct type of fish feed. Short term shortages of fish feed limits the growth of the salmon, and shortages thus have to considered as either in-feasible solutions or have to be modelled with a penalty cost similar to the potential revenue lost. External deliveries will be allowed to increase the feasibility of the problem.

2.3 Scope and Limitations

The task includes finding a method for providing a plan for feed transportation from a fish feed factory to fish farming locations at the Norwegian coast. The decisions that need to be made include the routing of the ships, a time schedule for different routes and amount of cargoes loaded and unloaded at each port.

As the problem is highly complex and is prone to great uncertainties, simplifications will be made in order to formulate problems that can be solved in a time frame that is suitable for everyday operations of the feed distribution system. This means that parameters as feed consumption will be modelled as constant throughout a given planning period. Travelling times between fish farms will be considered fixed for the given vessels and loading and discharge rates will be considered fixed. Heuristics and stochastic optimization will be considered in order to solve the problem within reasonable time frames. The solutions found by these optimization methods may not be equal to the global optimum solution, but will usually provide significant improvements to operations, relative to a schedule designed solely by intuition and simple hand calculations.

Only the distribution system itself and feed consumption will be considered in the optimization process. The distribution system includes loading and discharge from ships to ports and the sailing between locations. Feed production and earlier stages of the supply chain will not be considered, and production rates will be considered as deterministic.

3 Fish Feed Manufacturing and Distribution

This section will give a brief introduction to the salmon farming industry, with the main focus being on the feed manufacturing and distribution part of the industry. Section 3.1 will give an overview on the largest players in the salmon farming industry. Section 3.2 includes the production stages for salmon farming, from fertilization of eggs to the harvesting of the mature salmon. Section 3.3 gives an introduction to feed consumption, with data and estimates for feed conversion rates and deliveries. Section 3.4 lists the largest fish feed producers in Norway and the developments in feed production, while section 3.5 is centered on the challenges of fish feed distribution.

3.1 Salmon Farming Entities

Salmon farming on an industrial scale was developed in the 1970s, and the industry have seen an exponential growth since the first salmon farming locations first started operating. Due to the salmon farming operations being limited to areas with stable and cold sea temperatures, only a handful countries have been able to capitalize on the industry. As of 2018 the largest producers of Atlantic salmon included Norway, Chile, the United Kingdom, Canada and the Faroe Islands.

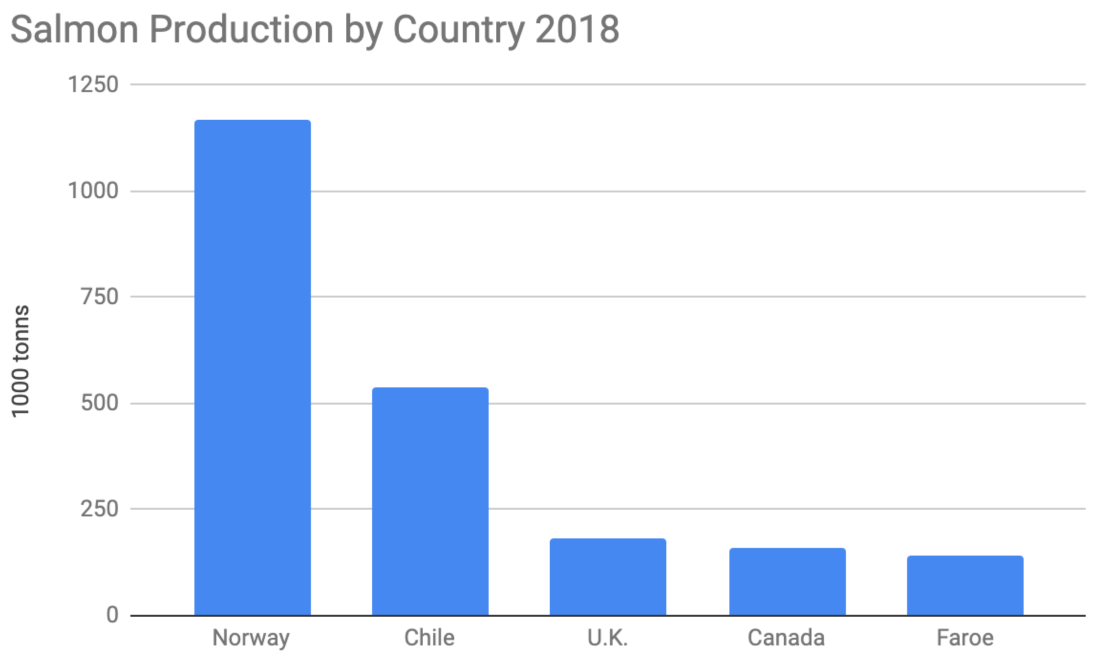

Figure 1: Production of farmed Atlantic Salmon by country. Plotted with data from FishChoice Inc. (2018).

The salmon farming industry have traditionally been fragmented with a large number of companies. However, recently a consolidation process have lead to many smaller companies either being bought out or merged to larger entities. Mowi have been among the leading companies in this consolidation process, by takeovers of Hydro Seafood in 2000 and Fjord Seafood, Stolt Sea Farm and Pan Fish in 2006. Together with Lerøy, SalMar and Cermaq, they now constitute the group of largest salmon farming companies in the world. They are all based in Norway. In Chile, the largest producers of Atlantic Salmon includes AquaChile, Pesquera Los Fiordos and Salmones Camanchacas. Cooke Aquaculture holds a leading position in Canada, while Bakkafrost remains the dominant player at the Faroe Islands.

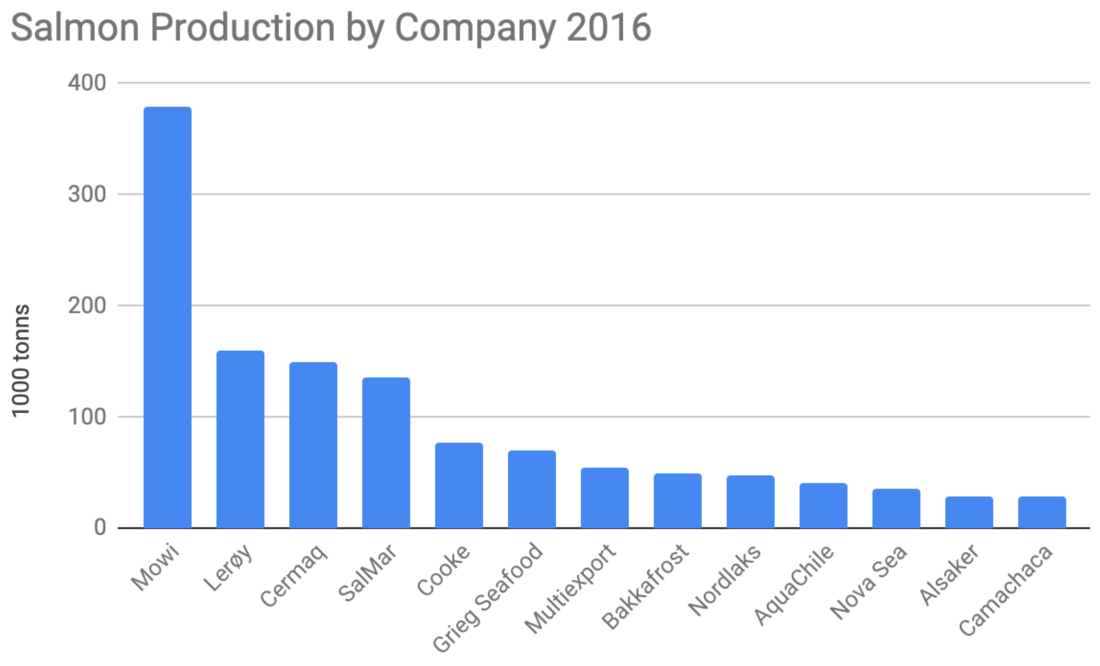

Figure 2: Production of farmed Atlantic Salmon by company. Plotted with data from Berge (2017).

3.2 Production Stages of Salmon Farming

Production of farmed salmon is a multi-stage process, where the two main stages can be divided into freshwater and seawater production. The total cycle from hatching of eggs to the slaughtering process takes 2 to 4 years.

The fish spends 10 to 18 months in freshwater(Melberg and Davidrajuh, 2009). Milt from mature male salmon are used in order to fertilize eggs from female salmon. The eggs are kept in incubation tanks with about 5 000 eggs per liter of water. The eggs will hatch after 490 degree-days. This implies that if water is kept at 10 degrees Celsius, it takes 49 days to hatch the eggs. When eggs hatch, the fish is called an alevin. The alevin carries a yolk-sac with sufficient amounts of nutrition until the alevin have grown large enough to consume normal feed by itself. The salmon, now referred to as fry, is around 290 degree-days when this happens. Fry is usually around 25 mm large and placed in tanks with flowing water and subdued lighting. Fry later grows into parr. In the parr stage the fish grows fast, and is placed in larger freshwater tanks. Before the salmon is ready to enter seawater it has to go through a smoltification proecess. Smoltification starts when the fish is around 60 grams and 120 mm. Fish farmers want to utilize the biomass licences optimally, and ideally transfer smolt to seawater multiple times a year. The smoltification process is therefor sometimes delayed by lowering temperature and light. Parr that become smolt their second spring are referred to as S1, while parr that takes one extra year is referred to as S2. By increasing light and temperature, smolt can be ready by autumn, 6 months before normal (Sigholt et al., 1998). These smolt are referred to as S0.

Before smolts are transferred to seawater they are placed in tanks with gradually increasing salinity. The smolts are then transferred by live fish carriers, also referred to as well boats, to their farming locations at sea. The fish grows rapidly in seawater and usually reaches a weight of 4.5 to 5.5 kilos in a period of 12 to 18 months. When fish reach sexual maturity, the fish stops growing and instead starts producing eggs and milt. It is therefore economically beneficial to harvest salmon before they reach this stage.

Fish are usually harvested when they reach a weight of 5 kilograms. While weight and fish reaching sexual maturity are clear indications that fish should be harvested, other factors may also impact the timing of harvesting. These factors include salmon pricing, and the premiums for certain weight classes for salmon. Also threatening diseases or algae can be deciding factors when evaluating the timing of harvesting. Production sites are fallowed after a production cycle, meaning that it is kept empty for at least two months between production cycles. This is done in order to prevent the spreading of lice and threatening diseases between different generations of farmed salmon.

3.3 Feed consumption

The feed needed at a given fish farming location can be estimated with feed tables (Juell et al., 1993). The growth rate for the fish is usually distinguished from the feed conversion ratio. The multiple of these two factors, will give the feed needed for a given biomass. The consumption of fish feed will vary based on the size of the fish and the temperature in the water. Also other factors as lice and general fish health will impact consumption of food. This section will address the major impacts of feed consumption.

3.3.1 Feed conversion ratio

Salmon is cold-blooded, meaning that the temperature of its body varies with the temperature in the sea. Thus, less energy is needed in order to generate heat. This results in a low food consumption, relative to other animals as poultry and cattle. Estimates vary a bit between different sources, but about 1.3 kilos of fish pellets feed is needed in order to produce 1 kilo of salmon on average (Ellingsen and Aanondsen, 2006). This number will also vary based on other factors, such as fish size, water temperature, environment and general fish health.

The estimated average feed conversion ratio for farmed salmon in Norway is plotted in figure 3, with data provided by Kontali Analyse (2018). The feed conversion ratio is given for each month of the year.

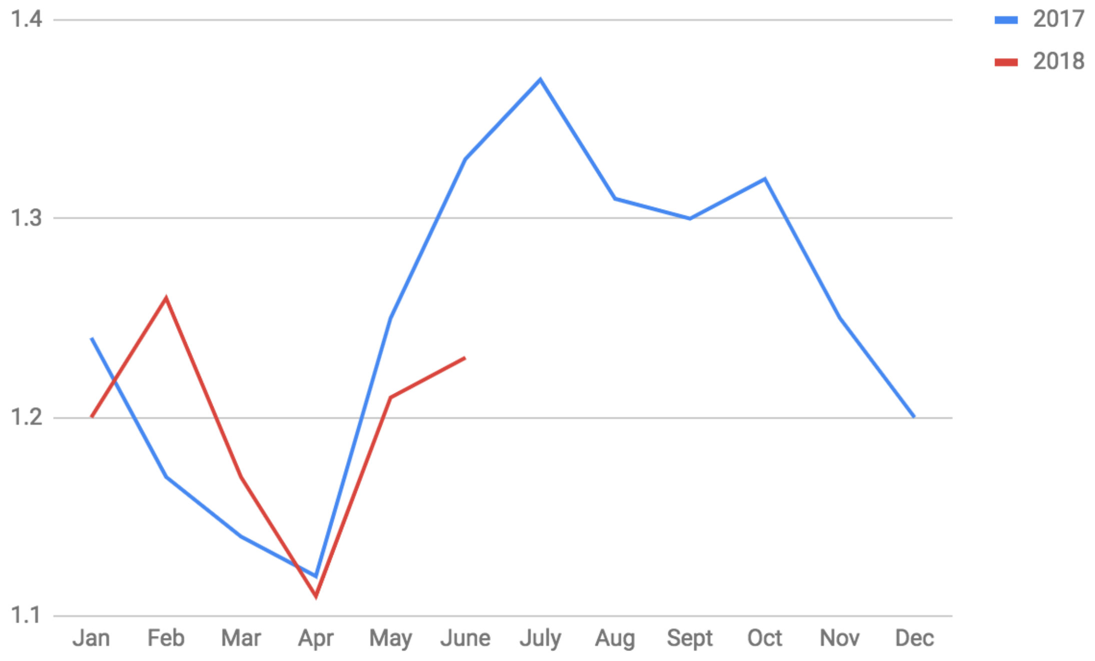

Figure 3: Feed Conversion Ratio estimates for Norwegian salmon. Plotted with data from Kontali Analyse (2018).

3.3.2 Fish growth rates and consumption

Fish growth rates are also dependent on many of the same factors as the feed conversion ratio. Abiotic factors includes light and temperature (Austreng et al., 1987), while biotic factors, as fish size, are also important (Brett et al., 1979). Local factors as environment and nutrition can also impact growth (Fivelstad et al., 2007).

The optimal temperature for growing salmon is around 8 to 14 degrees Celsius. In general, feed consumption increases with warmer weather. Compared to other salmon farming countries, Norway have larger underlying variations in temperature, thus also larger changes in feed consumption.

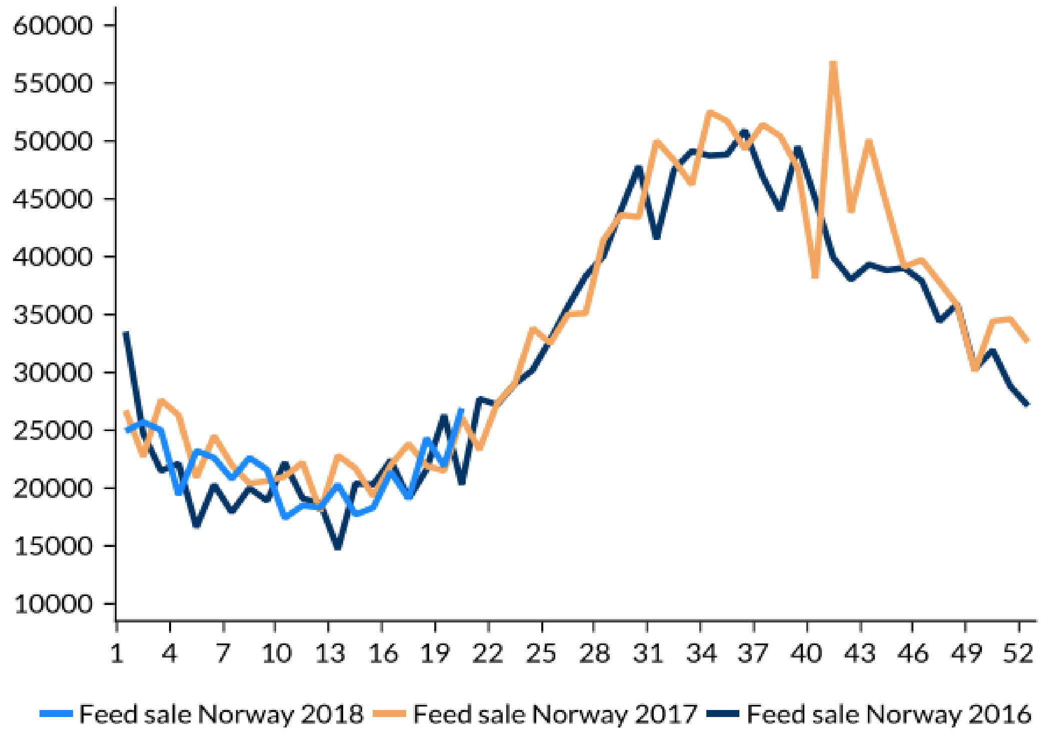

Figure 4: Feed Consumption estimates for Norwegian salmon farming, measured in tonnes per week. Source: Kontali Analyse (2018)

As figure 4 shows, the minimum feed consumption is only about 30% of the maximum feed consumption. Considering this is the combined consumption for Norway as a whole, local changes may even be more severe. Thus it would be inefficient to rely on a fixed schedule for feed delivery throughout the year.

Requirements of fish feed will also be impacted by other factors. Where the fish is in its life cycle will impact both the type of feed and the amount consumed. Consumption of feed tends to be higher at the end of the life cycle. Also singular events may have severe implications for feed requirements, such as lice and other illnesses that require medication or even immediate harvesting or starving of the fish. Mass-escapes of fish could also impact the feed consumption significantly.

Meal feeding and continuous feeding are the two main ways of feeding the fish. Meal feeding is based on feeding the fish in larger doses a given number of times a day, while continuous feeding gives the fish a constant flow of feed throughout the day. Changes in the feeding strategy is based on sea temperature, fish size and the type of feed used.

3.4 Production of fish feed

There are 4 major players in the fish feed market in Norway. These include EWOS (Cermaq, 2016), Biomar (Aktieselskabet Schouw & Co., 2018), Skretting and Mowi (EY, 2017). Being the largest producer of farmed salmon in Norway, Mowi to a large extent supplies their fish farming locations with feed from their own factory at Valneset, Bjugn. The factory is capable of producing 400 000 tonnes of fish feed a year (Marine Harvest, 2018a). Except for downtime during maintenance, it is operating continuously throughout the year at all times. The inventory capacities at the factory include 26 000 tonnes of raw materials, split between 16 000 tonnes of dry raw materials and 10 000 tonnes of wet raw materials. Storage also includes 10 000 tonnes for finished fish pellets feed. The size of the pellets vary between 0.6 millimeters and 13 millimeters in diameter, depending on the size of the fish.

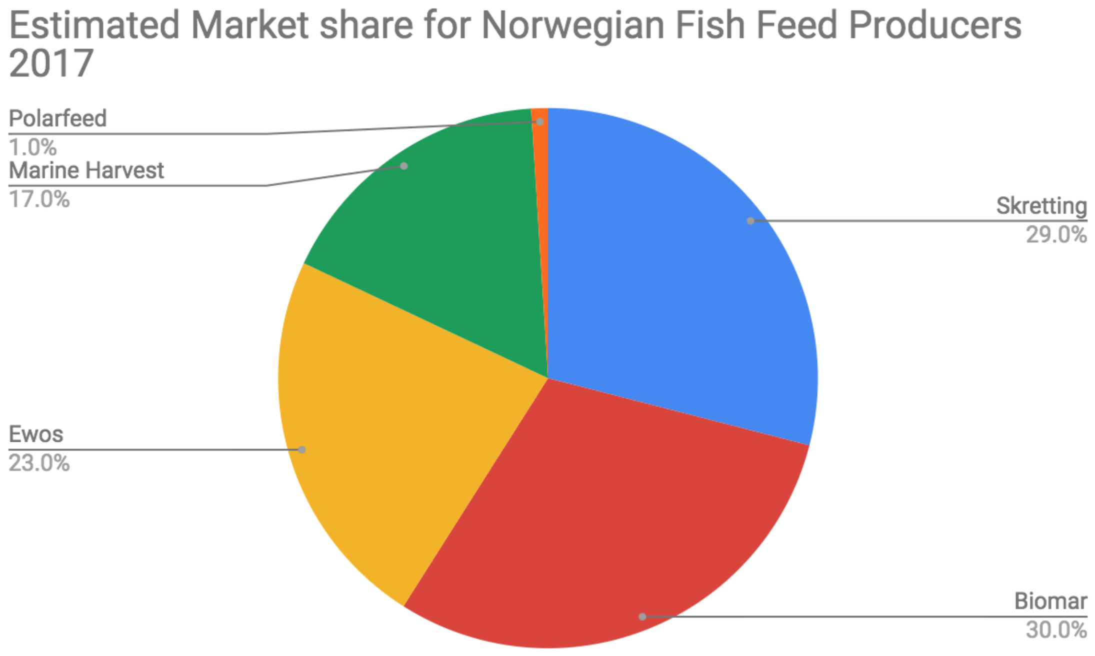

Figure 5: Estimated market share for fish feed producers in Norway for 2017. Plotted with data from Marine Harvest (2018a).

The process of making the fish feed starts with raw materials. The raw materials used are divided into macro materials, micro materials, and oils. Macro materials include fish meal, grain and plant materials. Micro materials include vitamins and minerals. Under production the different raw materials are weighted in order to find the optimal proportions, and then mixed before water is added, and the pellets are extruded to their desired size and shape. The pellets are then treated with oil in order to obtain their desired characteristics.

The methods of making fish feed have developed rapidly the last few decades. Going back to 1970's, smaller farmers mixed their own feed made with fish waste, fish meal and shrimp-shell. Today a more knowledge-based process is used in order to optimize profit for farmers and increase fish health. This is done by making feed appropriate for different stages in the salmons growth cycle, with a combination that ensures that correct and sufficient nutrients are provided at the right time. Trimmings from the traditional fishing industries and pelagic fish are used in order to make fish meal and fish oil. As this process in not very effective, providing only 3-5% fish oil and 20-25% fish meal per kilo fish processed, a larger amount of vegetarian materials have been used.

As can be seen from the figures on the next page, the amount of vegetables in salmon feed have increased from 11% in 1990, up to 74% in 2014 (Ytrestøyl et al., 2014). The amount of fish oil used have decreased from 24% to 9 % in the same period, while the use of fish meal is down from 65% to 17%.

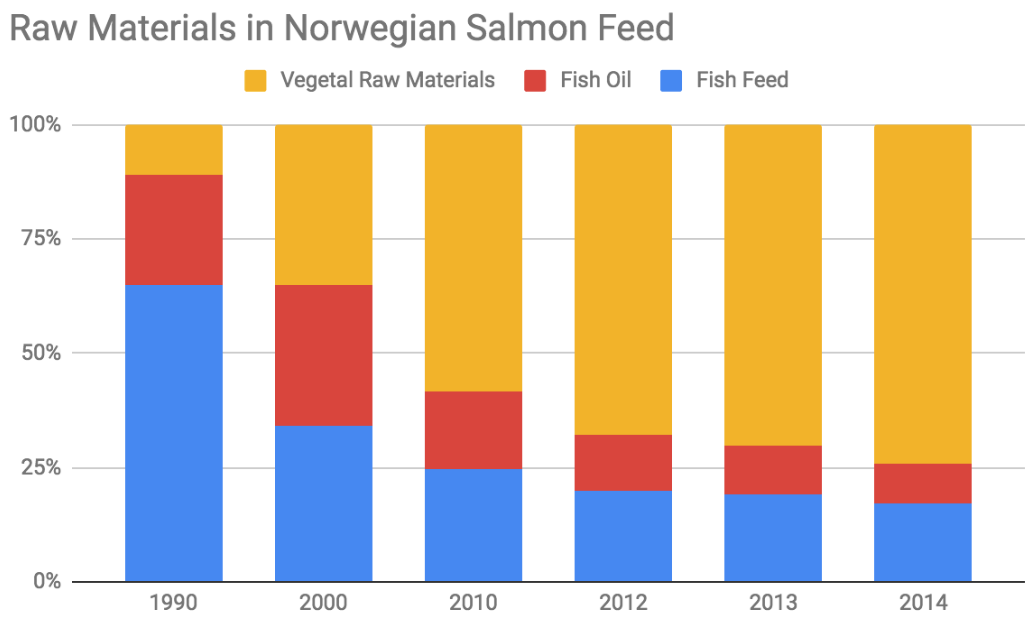

Figure 6: Development in raw materials used in production of fish feed. Plotted with data from Ytrestøyl et al. (2014).

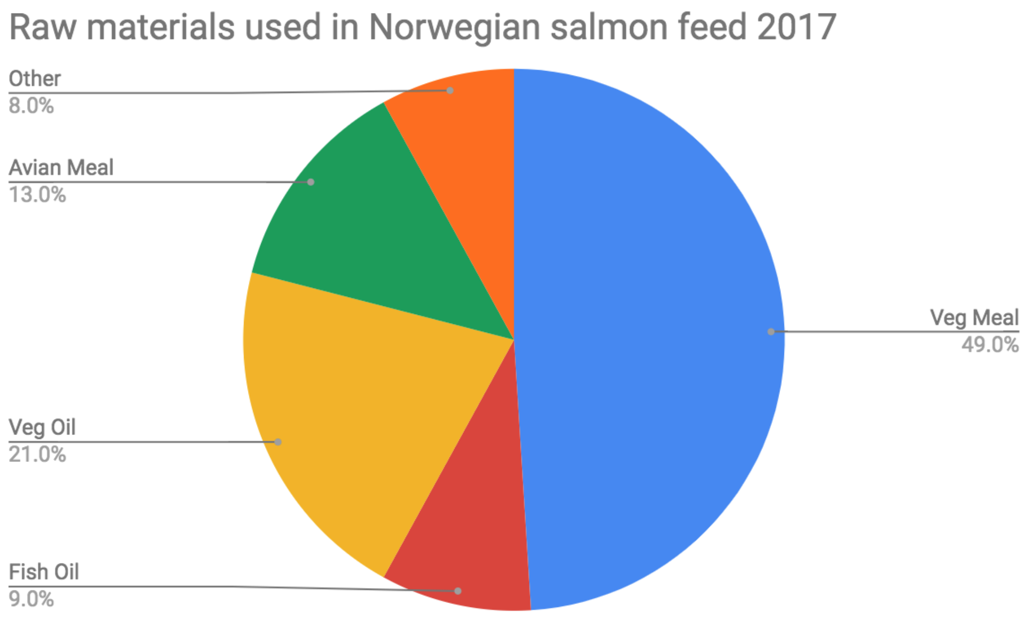

Figure 7: Raw materials used in production of fish feed in Norway in 2017. Plotted with data from Marine Harvest (2018a).

3.5 Distribution of Fish Feed

Distribution of fish feed is usually done by transport on ships called fish pellets carriers. There are two main types of fish pellets carriers used in these operations. These include ships carrying big bags with feed and ships with silo-like compartments filled directly with fish feed in bulk form.

Transport of fish feed leads to degradation of some of the feed by erosion. This occurs for feed both in bulk form and in big bags. Mowi assumes that about 0.5% of feed is lost due to degradation in bulk transport. In big bags the degradation is assumed to be even higher, and the degradation rate is significantly higher in the lower levels of the bags, than middle and lower levels. According to Aas et al. (2011), the degradation happened at a rate of 0.3% for the top levels of the feed in big bags, while to middle layers saw degradation rates of 0.8% on average for three different feed types. This is troublesome, as pulverized feed leads to a decreasing water quality and and overgrowth of algae, impacting the health and growth rates of the fish.

Mowi currently have 4 fish pellets carriers in operation. These vary in terms of cargo capacities and speed, as well as delivery capabilities. The cargo capacity limits how much feed the vessel can carry, while the speed limits the number of fish farms that can be served on a given route and the geographical locations of the fish farms server. Fish farming locations in narrow fjords and shallow waters can also prohibit larger vessels from serving these locations. As some fish farming location are only able to handle big bags, while others are only able to handle fish feed in bulk, there are clear limitations determining which vessels can serve which farms based on the delivery system used. This increases the perceived complexity of the problem, but can also be used in order to limit the number of variables in the problem.

The fish feed distribution can be also be impacted by a number of factors. Hash weather conditions can impact operations as wind and waves impact vessel speed and unloading operations. In extreme weather, discharge operations may not be possible at all, at certain locations.

If a salmon farming companies current fleet is not sufficient in order to keep feed inventory at wanted levels, external vessels can be hired. Vessels are often owned or operated by fish feed producers and some will require that the feed is also bought by from the fish feed company. The decision of hiring external vessels usually have to be taken weeks in advance. This complicates the planing, as the decision have to be made while there are still uncertainties for what the consumption rates and the inventories will be at the start of the planning period where external vessels are used.

4 Literature Study

4.1 Similar problems

This thesis builds on the work done by Haugland and Thygesen (2017) and their paper "Use of Clusters in a Route Generation Heuristic for Distribution of Fish Feed". The aim of this report is to further develop their methods for optimization and clustering.

Haugland and Thygesen have done substantial work related to the modeling of the problem and finding good ways to cluster nodes. Their model is based on an VRPMT, and solved using a route generation method. The main deficiencies with this model include the lack of time windows and the lack of constraints with regards to the inventory levels. In the given model each fish farming location may be visited multiple times, however there are no constraints that prohibits a given location to be visited two times in a row. In reality restrictions with respect to maximum inventory levels may prohibit this. Also, the only way to account for the current inventory levels at locations in their models, is by manually changing the demand at each location. The main benefit includes the reduced complexity that the VRP solved using the route generation method offers.

4.2 VRP - Vehicle Routing Problem

The vehicle routing problem describes how a set of vehicles with given capacities may be utilized in order to serve costumers. The objective is to reduce the total cost of serving the costumers. The problem was first described by Dantzig and Ramser (1959) and was focused on petrol deliveries. As the VRP is an NP-hard problem, heuristics are often used in commercial solvers (Toth and Vigo, 2002). In the classical form of the VRP each costumer belongs to exactly one route. The problem does not consider a finite number of vehicles, time horizons or the assignment of a given route to a specific vehicle. This can be accounted for by reformulating the problem into a Multiple Trip Vehicle Routing Problem (VRPMT).

The VRPMT accounts for a finite number of vehicles and time horizons. In addition one may utilize the same vehicle on multiple routes. Fagerholt (1999) discusses transportation of goods from ports in Norway to Europe and the US, and formulates the problem as a VRPMT. Fagerholt uses a route generation method, that first generates feasible routes, and then optimizes which routes to be selected. The paper also discusses the use of multiple depots.

4.3 IRP - Inventory Routing Problem

The inventory routing problem combines inventory management with vehicle routing, in order to serve consumers at different geographic locations and minimize cost. An integrated solution is provided as inventory management, vehicle routing, and delivery scheduling are simultaneously optimized. Attributes of IRP models introduced in relevant papers can be seen in table 1.

Paper	Fleet	Products	Compartments	Time	Stochastic	Network
Bell et al. (1983)	Heterogeneous	Single	Single	Continuous	Deterministic	
Campbell et al. (1998)	Homogeneous	Single	Single	Continuous	Stochastic	
Christiansen (1999)	Heterogeneous	Single	Single	Continuous	Deterministic	Many-to-many
Agra et al. (2013)	Heterogeneous	Multiple	Multiple	Continuous and Discrete	Deterministic	Many-to-many
Nurminarsih et al. (2015)	Heterogeneous	Multiple	Multiple	Continuous	Deterministic	Many-to-many
Santosa et al. (2016)	Heterogeneous	Multiple	Multiple	Continuous	Deterministic	Many-to-many
Agra et al. (2017)	Heterogeneous	Single	Single	Continuous	Deterministic	Many-to-many

Table 1: Attributes of IRP models formulated in related research.

Coelho et al. (2013) provides a review of the developments and literature on the IRP. The article describes different variants and extensions of the IRP, and how heuristics and metaheuristics have been used in order to solve IRPs. The article also describe the initial use of IRP in inventory management of industrial gases, and especially highlights the applications of the IRP in the maritime sector. Bell et al. (1983), Campbell et al. (1998) and Christiansen (1999) are among the articles mentioned here. Reviews of said articles, as well as reviews of work based on these articles, are following.

Bell et al. (1983) were the first ones to formulate the Inventory Routing Problem. The paper ”Improving the Distribution of Industrial Gases with an On-line Computerized Routing and Scheduling Optimizer” is centered on inventory management of industrial gases on costumer locations. The problem is considered as very complex and consists of up to 800 000 variables and 200 000 constraints. It is solved to near optimality, using a sophisticated Lagrangian relaxation algorithms.

Campbell et al. (1998) discusses the IRP in an article published in 1998. The article focuses on vendor managed resupply, and names the petrochemical and industrial gas industry as examples were this occurs as an emerging trend. The article focuses on how to structure the problem and possible ways to solve a given IRP. The article also discusses how a dynamic programming can be used in order to solve a Stochastic Inventory Routing Problem (SIRP) by measuring the inventory levels at the start of each day and then use a given strategy to route vehicles in the optimal way to resupply nodes. The implications of lacking or unknown probability distributions and neglected measurement costs are discussed, as well typical models lack of adjustments for seasonality, weekdays and other parameters affecting the systems expected performance. As discussed in section 4.1, there are a number of factors that can lead to changes in feed consumption and sailing speed in the feed distribution problem as well. Thus, being aware of stochastic factors is essential when modeling the problem.

When combined inventory and time constrained routing problems was considered in the paper Christiansen (1999), variants of the VRP was already widely adopted as a mean to solve marine routing problems. In order to better model transportation of ammonia from production ports to consumption ports, Christiansen introduces an IRP formulation. While also this formulation allows for minimized cost and time windows, it also accounts for inventory restrictions, production- and consumption rates in ports. Christiansen models the IRP as a continuous time problem, where consumption is considered fixed. The problem is solved by a Dantzig–Wolfe decomposition, first introduced in Dantzig and Wolfe (1960). The overall problem is solved using the Branch and Bound method. As the IRP used in this paper is adapted to maritime applications, the model is also relevant for the fish feed distribution problem.

Building on the work on IRP formulations in Christiansen (1999), the paper by Agra et al. (2013) provides a problem formulation centered on distribution of oil products to islands in the Cape Verde area. This short sea shipping case makes times in port significant and time windows are included in the problem formulation. A heterogeneous fleet of ships with different compartments are considered. A given compartment can only hold one product at the time. The problem relates to our problem as both problems considers nodes placed fairly close to each other, heterogeneous fleets and restrictions on inventory management at demand sites. In the article both a continuous time model with constant consumption rates and a discrete time model with varying consumption is considered. The results shows that the discrete time model provides better bounds, however the large number of variables involved in the mathematical model increases the computation time relative to the continuous time model. This article is also relevant for the fish feed distribution problem, as feed consumption varies with temperature and other factors. Thus, the implications of modeling the fish feed distribution problem as a discrete time problem, as opposed to a continuous time problem, have to be considered.

Nurminarsih et al. (2015) provides a report on a dynamic-inventory ship routing problem (D-ISRP) . They consider a heterogeneous fleet with consumption and production ports. Each ship can handle multiple different products, but the compatibility for different products vary, thus increasing the difficulty of the problem. The ships are structured with different compartments. They develop a new heuristic algorithm in order to solve the problem faster. The algorithm used consists of two parts, one initiation route part and one rerouting part. The algorithm generates interaction coefficients between ships and ports, in order to determine and improve a ships ability to serve a port and ports ability to increase total offloading for ships. The main benefit of modelling the problem as a D-ISRP as opposed to just a ISRP is that one can easily reroute ships if unexpected changes in inventory or a ships position occurs. The fish feed distribution problem is also prone to unexpected changes in inventory levels when consumption changes. Thus, the implications of a dynamic model should also be considered for the fish feed distribution problem.

The paper that corresponds the best with the problem we have at hand, is the "Combined ship routing and inventory management in the salmon farming industry" by Agra et al. (2017). The article is also focused on feed transport in the salmon farming industry. It assumes that 2 homogeneous fish pellets carriers are used in order to serve a given number of locations. 20, 40 and 60 locations are respectively used as inputs when solving the problem. The paper is relevant as it provides a model for distribution of fish feed, and also provides ways to tighten the constraints by adding sub-tour elimination constraints and dynamic cut generation of clique inequalities. It also provides metaheuristics that can be applied in order to solve the model in an efficient way.

4.4 Exact Solution Methods

4.4.1 Branch-and-bound

Land and Doig (1960) introduced the Branch-and-bound method. In this two-sided algorithm the branching part divides the problem into smaller sub-problems that are easier to solve. The bounding happens when a part of the solution is proven to be inferior to the optimal integer solution for another sub-problem, or the linear relaxation of the problem is infeasible. The branch-and-bound method can also be used in order to generate upper and lower bounds on the problem. The lower bound on a minimization problem is the lowest value solution found for a LP relaxation on a given level in a search tree, while the upper bound is the best integer solution found. The lower bound will normally increase when lower levels of the search tree is explored, as the feasible area of the linear relaxation is smaller. These bound are valid, even before the branch-and-bound algorithm terminates.

4.4.2 Cutting-plane method

Cutting-plane algorithms were first introduced by Gomory (1963). By solving the linear relaxation of a MILP, one is always guaranteed that there exists an extreme point or a corner point that is optimal, provided the feasible region is non-empty, doesn't contain a line and the LP has an optimum. If this point is not an integer solution, there must exist a linear inequality that separates the optimum from the convex hall of the true feasible region. By adding this inequality, called the cut, one is able to exclude the non-integer optimal solution from the feasible region. By repeating this process, the optimal solution can be found when the first optimal solution for the new feasible region with the cuts, is also a feasible integer solution. Cutting-plane algorithms may be used as a standalone algorithm, or as a part of other algorithms.

4.4.3 Branch-and-cut

Archetti et al. (2007) first introduced the branch-and-cut algorithm to solve IRPs with single vehicles. This branch-and-cut algorithm was further improved in Solyalı and Süral (2011), which uses a separation algorithm first proposed by Padberg and Rinaldi (1991). The separation algorithm was originally used for classical TSP problems. When a violated inequality in the TSP is found, the branch-and-cut algorithm checks weather constraints for the IRP are also violated, and when violated the inequality constraints are added to the LP relaxation and re-optimized. This is done for each node in the branch-and-bound tree. Coelho and Laporte (2014) generalizes this to a multi-vehicle problem by defining a solution improvement algorithm where vertices are removed and reinserted when a branch-and-cut algorithm finds new optimal solutions.

4.4.4 Dantzig-Wolfe decomposition

The Dantzig-Wolfe decomposition was introduced in Dantzig and Wolfe (1960). Most optimization problems with a large number of variables will lead to formulations with sparse matrices. The Dantzig-Wolfe decomposition utilized this property, by decomposing an LP problem into smaller sub-problems and leaving out certain sets of constrains. The sets of constraints that are left out constitutes what is called the master problem. By exploiting the convex combination properties of the LP problem, the problem is re-formulated into another linear problem, where the optimal point is a linear combination of the corner points obtained from the sub-problems.

The Dantzig-Wolfe decomposition was introduced in maritime IRPs by Christiansen (1999). In this paper the ship schedules and the visit sequences for harbours constitutes the sub-problems. Coupling constraints and auxiliary variables constitutes the master problem. Only ship routes and harbour visits leading to minimized costs in the master problem generated in the column generation step.

4.5 Heuristics

The development of heuristics used in IRPs are summarized in Bertazzi and Speranza (2012). For general routing problems, the development is described by the following order.

- Greedy heuristics
- Local search heuristics
- Metaheuristics
- Hybrid heuristics and matheuristics

The greedy heuristic is exemplified with the nearest neighbour heuristic and the cheapest insertion heuristic, both used for solving TSPs. The 2-opt heuristic is an example of a local search heuristic. Local search heuristics can be effective, but often end up in a local optimum that is not escaped. Genetic and evolutionary algorithms, tabo search and simulated annealing are example of metaheuristics used in routing problems. Lately, matheuristics have gained more traction. The matheuristics incorporate mathematical programming as a part of the heuristic. The matheuristics uses mathematical programming techniques in order to solve sub-problems, parts of an instances, restrict the search space and to explore neighborhoods that are promising.

4.5.1 Feasibility Pump heuristic

The feasibility pump heuristic was introduced in Fischetti et al. (2005), and is used to find initial feasible solution to generic mixed-integer problems. The heuristic works by generating a sequence of fractional solutions to a linear relaxation of a given problem, and then rounding the solution variables. The sequence of solutions are generated by minimizing a distance function representing the distance between the rounded value and the feasible region of the linear relaxation to the main problem. This is done until a feasible solution is found, or until the heuristic terminates after a given number of iterations or meets other stopping criteria. In the paper Fischetti et al. (2005), the heuristic is tested on 83 difficult mixed-integer problems and compared to the commercial solver ILOG-Cplex 8.1. The findings shows that the heuristic yields competitive results, and is unable to find feasible solutions for the root node for only 3 of the problems, compared to 19 for ILOG-Cplex 8.1.

In Fischetti and Salvagnin (2009) further improvements to the basic feasibility pump heuristic is proposed. The authors of the paper recognize the tendency to stall as one of the main drawbacks of the basic feasibility pump heuristic, leading to a high frequency of perturbations and restarts. The perturbations and restarts leads to oscillation in the distance function, instead of a smooth decline. This can be avoided by applying a smarter rounding method to the blind and simple rounding process used in the basic feasibility pump heuristic. Rounding based on constraint propagation is proposed as a way to do this. Constraint propagating is general method that consists of explicitly forbidding single values or combinations of values for variables of a problem. This logic is known as bound strengthening in integer optimization. By utilizing constraint propagation in the rounding phase, Fischetti and Salvagnin (2009) is able to better exploit information about the linear constraints.

Agra et al. (2014) combines the feasibility pump heuristic with a rolling horizon heuristics and a local branching heuristic in order to solve a maritime short sea IRP, where an oil company is responsible for the routing and scheduling of ships. By combining the three heuristics, integrality gaps are on average just half of what it is with just the rolling horizon heuristic.

4.5.2 Column generation Methods

Taillard (1999) solves a heterogeneous VRP by a heuristic column generation method. As efficient heuristics have already been developed for the homogeneous VRP, the solution of the heterogeneous VRP is achieved by solving a succession of homogeneous VRPs for all the different vessels. By combining tours generated by the different vessels with the homogeneous VRPs, a partial solution for the heterogeneous VRP is obtained. The partial solution is used as the starting point for a taboo search. This is repeated a number of times, and tours are memorized as candidates for a final heterogeneous VRP solution. A set T of tours are defined, and all tours visiting the same customers as another tour, but with higher costs are discarded. A set partitioning problem is then defined with the given tours remaining in set T. The set partitioning problem is then solved with a commercial solver, to obtain the final results.

4.5.3 Aggregation and Dis-Aggregation Methods

A heuristic approach with aggregation and dis-aggregation can be applied in order to solve the problem with a lower number of nodes initially. As the solution time of the problem grows exponentially with a increased number of nodes, clustering of the nodes can significantly improve solution times. After the aggregated problem is solved, a dis-aggregation heuristic have to be applied in order to obtain a solution including all customers. This solution can either be used directly or further improved. Newer versions of commercial solvers have implemented the possibility to insert starting values or hints for decision variables of the problem.

Aggregation and dis-aggregation techniques for optimization problems are discussed in Rogers et al. (1991). Model aggregation is applied in order increase tractability for large optimization problems. Aggregation and dis-aggregation techniques in mathematical programming consists of combining data, forming auxiliary model of reduces size and complexity and the analysis of results from the auxiliary model. Evans (1978) applies aggregation and dis-aggregation techniques in order to solve transportation problems with multiple products. The problems are solved to 80-85 % of optimality, with a computation time that is more than 1000 times faster than the full problem.

Weintraub et al. (1986) solves a mixed-integer linear problem related to forestry management by aggregating a larger problem. Timber stands are aggregated in macro stands and management alternatives are aggregated using weightings. The method used is first introduced in Zipkin (1980). More generally, sub-matrices are formed and multiplied with weighting vectors that consists of non-negative numbers that summarizes to one. Weintraub et al. (1986) proves that there must exist a weighting vector that ensures no loss in the objective value when the aggregation is done. Error bounds are also proposed. The aggregation method gives a 75% reduction in computation time for the forestry management problem, while error bounds ensures that the solution is within 7.7% of the objective value.

In integer programming the concept of surrogate constraints is an example of a group of aggregation and dis-aggregation strategies. Two or more constraints are multiplied with separate prime positive integers and then summarized, in order to aggregate the constraints. Elmaghraby and Wig (1970) utilizes this technique to solve linear integer problems for a stock cutting problem by sequentially adding all constraints into one single constraint.

4.5.4 Stochastic Optimization

Substantial research have been done on stochastic optimization algorithms. Craenen et al. (2001) discusses the use of evolutionary algorithms on constrained problems. As evolutionary problems often need a fitness function for each candidate, a candidate that doesn't meet the set constraints will need to be handled. Penalty functions can be implemented for violating constraint, however this does not always lead to desired results. In the paper, eliminating or repairing unfeasible candidates are suggested, as well as ways to transform the search space.

Evolutionary algorithms have also been applied to the IRP as shown in Othman et al.

(2016). Here the objective is set to minimize the inventory holding costs, transportation costs and shortage penalty costs. The constraints are implemented by defining a feasible region for the chromosome representation.

Evolutionary algorithms have also been combined with other stochastic optimization methods in order to solve IRPs, as shown in Santosa et al. (2016). The paper describes how a Inventory Ship Routing Problem (ISRP) with multiple products and a heterogeneous fleet and can be formulated. The problem that is formulated is NP-hard and a Cross-Entropy Genetic Algorithm (CEGA) is utilize in order to solve the formulated ISRP, claiming that it is faster than for instance a Tabu Search or exact methods. To speed up the computational time for a standalone Cross-Entropy (CE) method, it was combined with a Genetic Algorithm (GA), allowing for faster computations. Furthermore, the crossover mechanism in the GE was eliminated due to its complexity and lack of improved results.

4.5.5 Metaheuristic

In Dauzère-Pérès et al. (2007) the distribution of carbonate slurry to Europe with ships are discussed and formulated as an IRP. Slurries are transported from plants to tank farms. As only small instances could be solved with exact methods, a metaheuristic method was developed. The metaheuristic is comprised of multiple steps. The first one is based on a greedy algorithm, in order to find the transportation plan for each tank farm. Then a local search algorithm is used in order to swap tank farms. In addition, a genetic algorithm is used to explore the solution space. In the genetic algorithm, the chromosomes are representing the tank farm order.

The variable neighborhood search method is introduced in Mladenović and Hansen (1997). The algorithm is based on changing the neighborhood within a local search algorithm. A procedure is implemented in order to provide a new starting point within each neighborhood for the local search algorithm.

In Popović et al. (2012) a randomized variable neighborhood search heuristic is developed in order to solve an IRP related to multi-product fuel deliveries. The difference between the variable neighborhood search introduced in Mladenović and Hansen (1997) and the randomized variable neighborhood search, is that the neighborhoods are changed randomly. The results of the of the randomized variable neighborhood search showed better results than the deterministic CT heuristic.

4.5.6 Matheuristics

Bertazzi and Speranza (2012) cites multiple ways of applying matheuristics to IRPs.

- Minimizing the cost of routing

- A inventory first, routing second approach, where the inventory part of the decomposed problem is solved first and routes are found subsequently
- A cluster first, inventory routing second approach, where multiple customers are clustered into nodes first and the IRP is then solved for each node independently later
- A iterative customer based approach, where the optimal solution of the subproblem of a single customer is inserted into a partial solution
- A policy based approach, where the search space is limited
- A intensified tabu search approach, where tabu search scheme is applied

Yu et al. (2008) presents an approach of developing an approximate model for an IRP that is solved by applying a Lagrangian relaxation method. The problem is decomposed into two parts consisting of one inventory problem and one routing problem. The respective parts are solved by a linear programming algorithm and a minimum cost flow algorithm. A surrogate subgradient method is then used to solve the dual problem. Computational studies show that the given method can produce near optimal solutions for problems with up to 200 customers.

4.6 Parallel computing

As the applications for GPUs have become more widespread (Jung and O'Leary, 2006) with the rise of General-Purpose computing on Graphics Processing Units (GPGPU) and Artificial Intelligence (Lu et al., 2018), the interest in parallel computing have increased. While CPUs can be seen as sophisticated entities able to handle a wide range of tasks, the GPUs are more specialized and can only handle very specific problems. However, they are designed to handle a large number of these specific problems at the same time. This is due to the large number of Arithmetic Logic Units (ALU) working in parallel in the GPU (Fatahalian, 2010). Problems that can easily be separated into a number of parallel tasks is said to be "embarrassingly parallel". For a problem to be embarrassingly parallel, the solution of sub-problems must be independent of the result of other sub-problems. The main benefits with a GPU is that as long as a problem is embarrassingly parallel, the computational speed increases near linearly as more ALUs are added to the GPU.

Parallelism is not exclusive to GPUs, but can also be used in order to run problems on multiple cores in a CPU or even multiple CPUs. However, different aspect should be considered when optimizing code for CPUs. Findings show that while the main contributors to increased performance on a GPU includes minimizing global synchronization and utilizing local shared buffers, the main contributors to increased performance on a CPU are multithreading, cache blocking, and reorganization of memory accesses. (Lee et al., 2010)

Parallelism in branch-and-bound problems is discussed in Gendron and Crainic (1994). The paper discusses multiple aspects, including grain size, shared memory and communication, synchronous and asynchronous systems. Fine grain systems are systems consisting of smaller sub-problems, and they tend to be embarrassingly parallel to a higher extent than coarse grain systems. Fine grain systems typically scale to thousands of processors. While coarse grain systems used to be limited to a smaller amount of processors, there are now examples of being able to scale these more complex operations to thousands of processors as well. Synchronous systems uses the global clock in order to synchronize operations between processors, while asynchronous systems uses multiple clocks. 3 types of parallelism for the branch-and-bound algorithm are introduced. These include parallelism in operations on a generated sub-problem, parallelism by operations several sub-problems simultaneously and parallelism by operating on several branch-and-bound trees simultaneously.

5 Optimization Model

This section starts by discussing the possible applications for different model formulations, and the simplifications that can be made to these models. A total of three different optimization models are discussed in this section. The first one is a arc-load model and the second one is a arc-flow model. Possible formulation improvements are suggested for these models. At the end a comprehensive model is discussed. The comprehensive model is not considered further, but is relevant as the first two models can be seen as simplifications of the comprehensive model.

5.1 Applications and Simplifications

There are multiple ways to model the same problem. While problems can be modelled very precise, this often leads to a high degree of complexity with potentially many variables, many constrains and non-linear and in the worst cases also non-convex formulations. This leads to increased computational time, and in some cases we are not able to find the global optimum at all. This subsection will address the benefits and disadvantages of different ways to model the problem.

5.1.1 VRP or IRP

There are clear similarities with the VRP and the IRP. For short term planning the IRP can be formulated as an extension of the VRP. The complexity of the IRP increases for longer planning periods, but if solved optimally the use of IRP also has the potential to yield greater benefits for longer term planning. The main benefit with the IRP is that one always monitor the inventory levels at all nodes. The economical benefits of knowing that no fish farming locations will run out of feed is significant. The IRP also gives greater flexibility, as it only requires a ship to visits a given fish farm when inventory levels suggests so.

The main drawback with an IRP is the computational complexity. Another downside to the IRP is that it often tends to move as many deliveries as possible into the next planning period. Thus we may risk starting the next planning period with low inventory levels. This is called end of horizon effects. Schedules generated may also appear more random and less predictable from a human perspective, as they are a function of inventory levels, and not a fixed route.

By simplifying other factors and clustering nodes, the calculation times should still be within reasonable time frames. By adding restrictions to the problem for inventory levels at the end of the planning period, we are also able to limit the problem with deferrals to the next planning period, thus implementing a more long term approach to the planning. Another way of limiting the end of horizon effects is to impose larger minimum deliveries at ports.

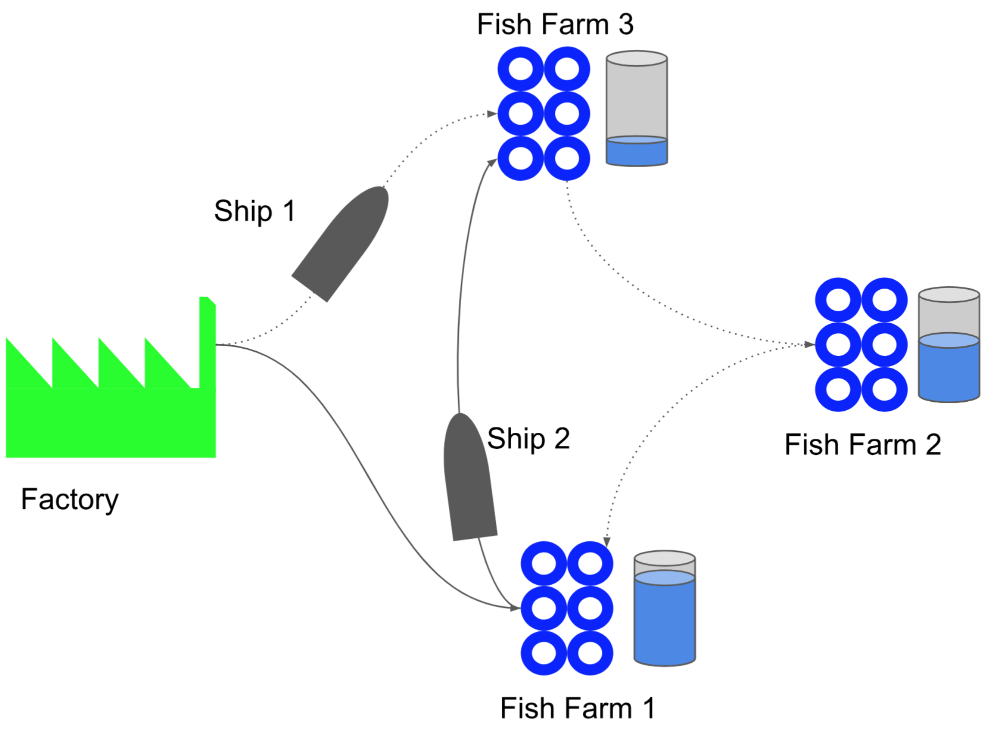

Figure 8: With an Inventory Routing Problem formulation, the model includes inventory management at all nodes, in addition to the ship routing. The figure shows how ships are directed in order to serve nodes before inventory levels of feed, illustrated with the cylindrical shapes, runs low.

5.1.2 Continuous Time or Discrete Time

Modelling the problem as a discrete time problem will give the opportunity to add variable consumption rates. This is beneficial in cases where we know that consumption of fish feed will be changed significantly during the planning period. It is already known that factors as temperatures and fish health plays an important part when determining the expected feed consumption. The downside to modelling the problem as a discrete time problem is the increased complexity and thus increased computation time.

By keeping planning periods fairly short, one is able to avoid most of large changes in sea temperature. It has therefore assumed that one can model the problem with constant rates of feed consumption.

5.1.3 Products and Compartments

It is known that different types of fish feed are produced based on factors as fish size, fish health and location. Modelling the problem with different products increases the number of variables in the problem and increases the complexity. The same holds true for compartments. While some fish pellets carriers carries bags of feed, others have different compartments that stores feed. It is possible to model these compartments, and the model can also implement the possibility for a compartment to only hold a given type of feed at a given time.

Fish pellets carriers with compartments, can have around 20 different compartments. It is therefore assumed that one is able separate different types of feed where that is needed. For fish pellets carriers with bags it is assumed that one is able to store different types of feed in different bags as needed. As route ahead is known when leaving the factory, it is assumed that one is also able to store bags in the correct order. Products and compartments have therefore not been included in the adjusted model.

5.1.4 Stochastic or Deterministic Model

As the feed delivery operations are prone to great uncertainties, like weather and waves, illness or escape of fish and consumption rates, formulation a stochastic model could potentially achieve operational benefits. However, adding stochastic parameters to a model would increase the complexity significantly. As a deterministic IRP model is already complex, the implementation of stochastic parameters would lower the number of nodes that could practically be implemented in the problem.

5.1.5 Clustering of Nodes

As the routing problem consists of around 150 different locations, the problem is extremely hard to solve to optimality. In order to simplify the problem, clustering of nodes can be done for farms located within a given geographical area. Clustering methods and application be found in section 7.

5.2 Arc-Load Formulation

The first model is a simplified arc-load formulation of IRP, such that only the most important factors are included. The model minimizes transportation costs, but also allows for external deliveries of feed, that adds extra feed and transportation costs. This is done while always ensuring that feed levels are above a predefined limit. The model holds the following properties:

- Inventory Routing Problem
- Deterministic Model
- Clustered Nodes
- Heterogeneous Fleet
- Single Product
- Single Compartment
- Continuous Time
- Constant Consumption Rates

5.2.1 Definitions

Set	Definition	Indexed By
H_T	Set of all harbours	i,j
V	Set of all vessels	v
H_v	Set of all harbours, can be visited by vessel v	i,j
M_{Ti}	Set of possible arrivals at port i	m,n
M_{iv}	Set of possible arrivals at port i by ship v	m,n
A_v	Set of all possible arches for vessel v	i,m,j,n

Varables	Definition	Unit	Variable Type
x_{imjnv}	1 if ship v routed directly form node (i,m) to (j,n), 0 otherwise	[-]	Binary
y_{im}	1 if node (i,m) not visited by any ship, 0 otherwise	[-]	Binary
z_{imv}	1 if route ends at node (i,m) for ship v, 0 otherwise	[-]	Binary
u_i	1 if cluster served by own vessels, 0 otherwise	[-]	Binary
q_{imv}	Quantity loaded or unloaded at node (i,m) by ship v	tonnes	Continuous
l_{imv}	Quantity on ship v after visiting node (i,m)	tonnes	Continuous
t_{im}^S	Start time loading at node (i,m)	hours	Continuous
t_{im}^E	End time loading at node (i,m)	hours	Continuous
s_{im}^S	Stock level at start of loading at node (i,m)	tonnes	Continuous
s_{im}^E	Stock level at end of loading at node (i,m)	tonnes	Continuous

Parameters	Definition	Unit
C_{ijv}	Cost of sailing arc (i,j) with vessel v	USD
E	Extra cost for feed with external delivery	USD/tonn
E_i	Transportation cost for external deliveries to node i	USD
W_{imv}	1 if ship v starts at port i at arrival m, 0 otherwise	[-]
J_i	1 if i is a load harbour, -1 if discharge	[-]
Q_{imv}^{MAX}	Upper load limit	tonnes
Q_{im}^{MIN}	Lower load limit	tonnes
C_v^{AP}	Capacity vessel v	tonnes
T	Planning period	hours
T_i^Q	Time to unload one unit at port i	tonnes/hour
T_{ijv}^S	Time to sail from i to j with ship v	hours
T_i^B	Minimum time from departure till next arrival at port i	hours
T_{im}^{WS}	Start of time window for arrival m at port i	hours
T_{im}^{WE}	End of time window for arrival m at port i	hours
R_i	Production rate (Negative if consumption)	tonnes/hour
S_i^{MIN}	Minimum inventory level	tonnes
S_i^{EMIN}	Minimum inventory level at end of planning period	tonnes
S_i^{MAX}	Maximum inventory level	tonnes
S_i^S	Stock levels at node i at start	tonnes

5.2.2 Objective

$$minz = \sum_{v \in V} \sum_{A_v \in (i,m,j,n)} C_{ijv} x_{imjnv} + T \cdot E \sum_{i \in H_T} R_i(1-u_i) + \sum_{i \in H_T} E_i^T R_i (1-u_i) \quad (1)$$

5.2.3 Network constraints

$$W_{imv} + \sum_{j \in H_v} \sum_{n \in M_{jv}} x_{jnimv} - \sum_{j \in H_v} \sum_{n \in M_{jv}} x_{imjnv} - z_{imv} = 0, \forall v \in V, i \in H_v, m \in M_{iv} \quad (2)$$

$$\sum_{v \in V} \sum_{j \in H_v} \sum_{n \in M_{jv}} x_{jnimv} + y_{im} + \sum_{v \in V} W_{imv} = 1, \forall i \in H_T, m \in M_{Ti} \quad (3)$$

$$y_{im} - y_{i(m-1)} \geq 0, \forall i \in H_T, m \in M_{Ti} \quad (4)$$

5.2.4 Loading and unloading constraints

$$x_{imjnv}(l_{imv} + J_j q_{jnv} - l_{jnv}) = 0, \forall v \in V, (i,m,j,n) \in A_v \quad (5)$$

Equation 5 can be linearized as shown in equation 6 and 7:

$$l_{imv} + J_j q_{jnv} - l_{jnv} + C_v^{AP} x_{imjnv} \leq C_v^{AP}, \forall v \in V, (i,m,j,n) \in A_v \quad (6)$$

$$l_{imv} + J_j q_{jnv} - l_{jnv} - C_v^{AP} x_{imjnv} \geq -C_v^{AP}, \forall v \in V, (i,m,j,n) \in A_v \quad (7)$$

$$l_{imv} - \sum_{j \in H_v} \sum_{n \in M_{jv}} C_v^{AP} x_{jnimv} \leq 0, \forall v \in V, i \in H_v, m \in M_{iv} \quad (8)$$

$$q_{imv} - \sum_{j \in H_v} \sum_{n \in M_{jv}} Q_{imv}^{MAX} x_{jnimv} \leq W_{imv} Q_{imv}^{MAX}, \forall v \in V, i \in H_v, m \in M_{iv} \quad (9)$$

$$\sum_{v \in V} q_{imv} + Q_{im}^{MIN} y_{im} \geq Q_{im}^{MIN}, \forall i \in H_T, m \in M_{Ti} \quad (10)$$

5.2.5 Time constraints

$$t_{im}^S + \sum_{v \in V} T_i^Q q_{imv} - t_{im}^E = 0, \forall i \in H_T, m \in M_{Ti} \quad (11)$$

$$x_{imjnv}(t_{im}^E + T_{ijv}^S - t_{jn}) \leq 0, \forall v \in Vm(i,m,j,n)\ inA_v \quad (12)$$

Equation 12 can be linearized as shown in equation 13

$$(t_{im}^E + T_{ijv}^S - t_{jn}) + M x_{imjnv} \leq M, \forall v \in Vm(i,m,j,n)\ inA_v \quad (13)$$

$$t_{im}^S - t_{Ei(m-1)} + T_i^B y_{im} \geq T_i^B, \forall i \in H_T, m \in M_{Ti} \quad (14)$$

$$T_{im}^{WS} \leq t_{im}^S \leq T_{im}^{WE}, \forall i \in H_T, m \in M_{Ti} \quad (15)$$

5.2.6 Inventory constraints

$$s_{im}^{S} - \sum_{v \in V} J_i q_{imv} + R_i t_{im}^{E} - R_i t_{im}^{S} - s_{im}^{E} = 0, \forall i \in H_T, m \in M_{Ti} \tag{16}$$

$$s_{Ei(m-1)} + R_i t_{im}^{S} - R_i t_{Ei(m-1)} - s_{im}^{S} = 0, \forall i \in H_T, m \in M_{Ti} \tag{17}$$

$$S_i^{MIN} u_i \leq s_{im}^{S} \leq S_i^{MAX}, \forall i \in H_T, m \in M_{Ti} \tag{18}$$

$$S_i^{MIN} u_i \leq s_{im}^{E} \leq S_i^{MAX}, \forall i \in H_T, m \in M_{Ti} \tag{19}$$

$$S_i^{EMIN} u_i \leq s_{im}^{E} + R_i(T u_i - t_{im}^{E}) \leq S_i^{MAX}, \forall i \in H_T, m \in |M_{Ti}| \tag{20}$$

$$S_i^{S} + R_i(t_{i1}^{E}) = s_{i1}^{S}, \forall i \in H_T \tag{21}$$

5.2.7 Variable constraints

$$x_{imjnv} \in \{0,1\}, \forall v \in V, (i,m,j,n) \in A_v \tag{22}$$

$$y_{imv} \in \{0,1\}, \forall v \in V, i \in H_v, m \in M_{iv} \tag{23}$$

$$z_{imv} \in \{0,1\}, \forall v \in V, i \in H_v, m \in M_{iv} \tag{24}$$

$$u_i \in \{0,1\}, \forall i \in H_T \tag{25}$$

$$q_{imv} \geq 0, \forall v \in V, i \in H_v, m \in M_{iv} \tag{26}$$

$$l_{imv} \geq 0, \forall v \in V, i \in H_v, m \in M_{iv} \tag{27}$$

$$t_{im}^{S} \geq 0, \forall i \in H_T, m \in M_{Ti} \tag{28}$$

$$t_{im}^{E} \geq 0, \forall i \in H_T, m \in M_{Ti} \tag{29}$$

$$s_{im}^{S} \geq 0, \forall i \in H_T, m \in M_{Ti} \tag{30}$$

$$s_{im}^{E} \geq 0, \forall i \in H_T, m \in M_{Ti} \tag{31}$$

5.2.8 Model Explanation

The objective function is structured in order to minimize the total costs of feed deliveries. The first term of the objective function adds the sailing costs between destinations. The second term adds the marginal cost if feed is from external sources. In other words, this is the difference between the cost of feed from the self-owned factory and the cost for feed at other sources. This includes all feed consumed at the given node during the planning period. The third term includes transportation costs for nodes served by external vessels.

The network constraints are defined in order to ensure that ships are generated, routed correctly and the eliminated. The constraint given by equation 2 ensures flow conservation, such that the numbers of vessels arriving and being generated at node, equals the number of vessels being eliminated and leaving the same node. Equation 3 ensures that a given harbour is only visited one time for each possible arrival slot. Equation 4 ensures that if an arrival slot is not used for a given harbour, the following arrival slots are not used either. This constraint is used in order to tighten the constraints.

Loading and unloading constraints are defined in order to ensure that all ships carry the correct load when arriving and leaving each port. Equation 5 ensures that vessel cargo is in accordance with the loading and unloading at ports visited. Equation 8 ensures that the maximum capacity of the ship is not surpassed. Equation 9 and 10 sets upper and lower bounds on how much feed that have to be unloaded during one port visit.

Time constraint are defined in order to ensure that sailing and loading happens in the correct order, with sufficient amounts of time dedicated for each task and waiting. Time constraints also generates a schedule for operations when solving the problem. Equation 11 ensures that the end of operations at a port for a given arrival happens at a time equal to the loading or discharge process time after the start of the process. Equation 12 ensures sailing routes and schedules comply. Equation 14 ensures that the start of a loading or unloading process at a port can not happen before the given process is finished for the previous arrival. Equation 15 ensures that operations happen within the given time windows.

The inventory constraints are defined in order to ensure that inventory levels are in accordance with deliveries and consumption rates, and within given bounds for inventory levels. Equation 16 ensures that inventory levels at the end of a loading or discharge process is correct with respect to the inventory level at the start of the process, consumption and deliveries. Equation 17 ensures correct changes in inventory levels at ports between arrivals. Equation 18 and 19 ensures that inventory levels are within predefined limits at arrivals, and equation 20 ensures that inventory levels are still within predefined levels at the end of the planning period.

5.3 Arc-Flow Formulation

An alternative way of linearizing constraint 5 can be achieved by reformulating the model from an arc-load model into an arc-flow model. Instead of assigning variables l_{imv} for the load onboard a ship when leaving a node, the arc-flow model uses variables l_{imjnv} for the load carried on each arc. By replacing constraint (5) - (8) with the following constraints, the model is reformulated into an arc-flow model.

$$\sum_{j \in H_v} \sum_{n \in M_{jv}} l_{jnimv} + J_i q_{imv} - \sum_{j \in H_v} \sum_{n \in M_{jv}} l_{imjnv} = 0, \forall v \in V, i \in H_v, m \in M_{iv} \tag{32}$$

$$l_{imjnv} \leq C_{APv} x_{imjnv}, \forall (i, m, j, n) \in A_v, v \in V \tag{33}$$

$$l_{imjnv} \geq 0, \forall (i, m, j, n) \in A_v, v \in V \tag{34}$$

Constraints 32 ensures that the flow on an arc is equal to the preceding arc for the same vessel, adjusted for the loading or discharge at the last port. Constraint 33 ensures that the flow on any given arc is 0 if the arc is not used, and always less than the capacity of the vessel used. Constraint 34 ensures non-negativity for all arc-flows.

5.4 Formulation Improvements

5.4.1 Tightening time windows

Time windows are already implemented in the problem formulation in section 5.2. However significant improvement in computation time could be achieved if these time windows were tightened. As most fish farms today are fully automated, not requiring additional crew present in order to handle fish feed deliveries, time windows are not relevant to meet crew requirements at most ports. Thus, imposing artificial time constraint could lead to worse solutions being generated.

However, without loss of possible feasible solutions the first arrivals for nodes can be set equal to the time it takes for the closest ship to travel to the given node. Subsequent visits are set to the earliest possible time found for the first arrival, plus the minimum time between visits. The latest possible first arrival at a fish farming node, is the moment when inventories run out. For subsequent arrivals at fish farming nodes, the latest possible arrival is the last departure time plus the time it takes to consume all feed. Similarly for the feed production facilities, the latest possible time for the first arrival is the moment when inventory levels are maxed out, and the latest possible time for subsequent arrivals are the last departure time, plus the time it takes to fill the inventory again.

5.4.2 Sub-tour elimination constraints

Sub-tour elimination constraint are used in Vehicle Routing Problems, but may also be used in Inventory Routing Problems in order to tighten constraints. Formulations of sub-tour elimination constrains for the fish feed distribution problem is defined as follows (Adulyasak et al., 2015).

$$\sum_{(i,m)\in N} \sum_{(j,n)\in N} x_{imjnv} \leq |N| - 1, N \subseteq H_v \times M_{iv}, v \in V \tag{35}$$

A stricter alternative can be formulated by aggregating the ships into one constraint.

$$\sum_{v\in V} \sum_{(i,m)\in N} \sum_{(j,n)\in N} x_{imjnv} \leq |N| - 1, N \subseteq H_T \times M_{Ti} \tag{36}$$

5.4.3 Tightening constraints by dynamic cut in branch-and-bound

Clique inequalities was introduced in Agra et al. (2014) for maritime applications. By applying clique inequalities we are able to exclude conflicting routing combinations. The following pairs of binary variables are defined such that only one variable in each pair can be set to 1 for each solution, while the remaining variable have to be set to 0. The conflicting routing combinations include the following:

x_{imjnv} and x_{jnimw} where $(i,m),(j,n) \in M_{Ti}, v, w \in V$
x_{imjnv} and x_{jmkow} where $(i,m),(j,n),(k,o) \in M_{Ti}, v, w \in V$
x_{jnimv} and x_{koimw} where $(i,m),(j,n),(k,o) \in M_{Ti}, v, w \in V$
x_{jnimv} and x_{imjow} where $(i,m),(j,n),(j,o) \in M_{Ti}|n > o, v, w \in V$
x_{jnimv} and x_{imkow} where $(i,m),(j,n),(k,o) \in M_{Ti}, v, w \in V|v \neq w$

The first conflict implies that if an arc is used, the arc representing the opposite way of traveling can not be used, as that would imply traveling back in time. This is because a arrival at a port is tied to a variable indicating a specific time. The second conflict implies that one can not go from one location to more than one other. The third conflict implies that you can only come from one location when you go to any other. The fourth conflict implies that you can not come to a location and then visit the same location with a lower arrival number. The fifth conflict implies that when arriving at node with a ship, the same ship will have to leave the node.

5.5 Comprehensive model

While no model is perfect, a comprehensive model is shown in Appendix A3. It includes the modelling of different products, compartments and compartment washing. With a high number of fish farming location, this model would consist of too many variables to be solved in a feasible time-frame. The model in Appendix A3 is left as an example of a formulation that may be solved in the future if computational resources is substantially increased, or more efficient heuristics are developed.

The optimization model in Appendix A3 is based on the IRP models of Santosa et al. (2016), but restrictions for inventory levels at the end of the planning period are added. The nature of the IRPs tens to defer as many deliveries as possible to the next planning period. Without modifications to the problem, one might end up with inventory levels running low, before one is able to serve all locations in the next planning period.

5.6 Discussion

The models defined in this section provides the fish farming company Mowi with a tool that can be utilized in order to optimize routing and scheduling of ships, while at the same time ensuring that inventory levels stays above a predefined limit. The value provided for the fish farming company is mainly related to the reduced fuel consumption brought by optimal routing of the ships and the non-zero inventory levels, that ensures continuous feeding and growth of fish.

Choosing to formulate the model as an IRP, as compared to a VRP increases the computational complexity significantly. This also increases the expected computational time, which is likely to reduce the practical applications of the model. In the fish farming industry, time is often a critical factor and unforeseen events may demand quick decision making. Thus a too complex model may be of little value.

In order to keep the complexity of the model at a reasonable level, simplifications as single compartment and single product implementation have been used. As the scheduling and routing are given by the model, the optimal loading of products in the different compartments of the ships can be carried out based on these results. As there are few restrictions on which compartment can handle which feed type, this is not likely to limit real life operations.

The most significant simplifications in order to reduce the complexity of the problem include the clustering of the nodes. The clustering of nodes reduces the number of possible arcs that can be utilized by the ships, and thus also reduces the number of variables in the problem.

As opposed to most other IRP models found in research literature, external deliveries are implemented as an option for each fish farm. Also independent constants are added for inventory level restrictions at the end of the planning period in order to avoid end of horizon effects. The consumption scenarios carried out in the computational study shows the economic benefits of using external vessels to serve remote locations.

6 Decomposition

Decomposition of optimization problems have been used in order to solve complex problems, by dividing them into sub-parts where only some of the constraints and variables are considered in each part. A Master problem is then solved with input from the already solved sub-problems.

Christiansen (1999) successfully applies a decomposition approach in order to solve an inventory routing problem of similar structure to the problem presented in this thesis. The main ideas for the decomposition is reused in this thesis.

6.1 Dantzig-Wolfe Decomposition Theory

A linear programming problem on the following form may be solved using Dantzig-Wolf decomposition.

$$min\ c_1'x_1 + c_2'x_2 \tag{37}$$

such that

$$L_1x_1 + L_2x_2 = b_0 \tag{38}$$

$$M_1x_1 = b_1 \tag{39}$$

$$M_2x_2 = b_2 \tag{40}$$

$$x_1, x_2 = b_2 \tag{41}$$

The problem can be reformulation, such that the new problem have fewer equality constraints, but more variables. We start by defining P_i in equation (42).

$$P_i = \{x_i \geq 0 | M_ix_i = b_i\}, i = 1, 2 \tag{42}$$

If P_1 and P_2 are non-empty, the problem can be rewritten as in shown in equation (43)-(45).

$$min\ c_1'x_1 + c_2'x_2 \tag{43}$$

such that

$$L_1x_1 + L_2x_2 = b_0 \tag{44}$$

$$x_1 \in P_1, x_2 \in P_2 \tag{45}$$

For i=1,2, we define $x_i^j, j \in J_i$ as the extreme points of P_i and $w_i^k, k \in K$ as the extreme rays of P_i. By the Resolution Theorem of Hermann Minkowski (Bertsimas and Tsitsiklis, 1997) it can be seen that any element x_i of P_i can be formulated as shown in equation (46)-(49).

$$x_i = \sum_{j \in J_i} \lambda_i^j x_i^j + \sum_{k \in K_i} \theta_i^k w_i^k \tag{46}$$

were

$$\sum_{j \in J_i} \lambda_i^j = 1, i = 1, 2 \tag{47}$$

$$\lambda_i^j \geq 0, i = 1, 2, \ j \in J_i \tag{48}$$

$$\theta_i^k \geq 0, i = 1, 2, \ k \in K_i \tag{49}$$

Thus the initial problem can be stated on the form shown in equation (50)-(55).

$$min \sum_{j \in J_1} \lambda_1^j c_1' x_1^j + \sum_{k \in K_1} \theta_1^k c_1' w_1^k + \sum_{j \in J_1} \lambda_2^j c_2' x_2^j + \sum_{k \in K_2} \theta_2^k c_2' w_2^k \tag{50}$$

such that

$$\sum_{j \in J_1} \lambda_1^j L_1' x_1^j + \sum_{k \in K_1} \theta_1^k L_1' w_1^k + \sum_{j \in J_1} \lambda_2^j L_2' x_2^j + \sum_{k \in K_2} \theta_2^k L_2' w_2^k = b_0 \tag{51}$$

$$\sum_{j \in J_1} \lambda_1^j = 1 \tag{52}$$

$$\sum_{j \in J_2} \lambda_2^j = 1 \tag{53}$$

$$\lambda_i^j \geq 0, \forall i, j \tag{54}$$

$$\theta_i^k \geq 0, \forall i, k \tag{55}$$

By defining the dual variable for the first equality constraints as q, and defining the dual variables associated with the convexity constraints as r_1 and r_2 respectively, the function for reduced cost can be found. As the cost coefficient of a variable λ_i^j is given by $c_i' x_i^j$, the reduced cost of the λ_i^j variables for the respective subproblems is given as follows:

$$c_1' x_1^j - q' L_1 x_1^j - r_1 \tag{56}$$

$$c_2'x_2^j - q'L_2x_2^j - r_2 \tag{57}$$

In a similar fashion the reduced cost for the θ_i^k variables respectively is given as follows:

$$c_1'w_1^j - q'L_1w_1^j \tag{58}$$

$$c_2'w_2^j - q'L_2w_2^j \tag{59}$$

It can be noted that the r_i variables are not included in these formulations for reduced costs as there are no convexity constraints for the θ_i^k variables.

Instead of evaluating the reduced cost for every variable, a linear programming problems can formulated on the form

$$min\ c_i'x_1^j - q'L_ix_i^j \tag{60}$$

such that

$$x_i \in P_i \tag{61}$$

for each of the subproblems. These may be solved by simplex or other optimization algorithms. Three different alternatives have to be considered.

- The optimal cost is $-\infty$. In this case the optimization will yield an extream ray w_i^k and the reduced cost of θ_i^k is negative, and the associated column will be generated.
- The optimal cost is finite and less than r_i. In this case the optimization will yield an extreme point x_i^j and the reduced cost of λ_i^j is negative, and the associated column will be generated.
- The optimal cost is finite and greater or equal to r_i. In this case the reduced cost for both the θ_i^k and λ_i^j variables are nonnegative.

If a variable of reduced cost is found, the associated column will enter the basis in the master problem, which is then resolved in order to generate new dual variables.

6.2 Model reformulation

The arc-load formulation presented in section 5.2 is used as a starting point for the decomposition. The arc-load problem can be divided into one sub-problem representing the harbour inventory problem and one sub-problem representing the ship routing problem. In addition to these sub-problems, a master problem have to be formulated in order to ensure that the solutions found for the sub-problems are compatible with each other and the coupling constraints defined in the original arc-load formulation.

When decomposing the arc-load problem, the problem does not fall apart naturally for all constraints and variables. The variable t^S_{im} is governing the loading or discharge time at a given harbour, while simultaneously being related to all the ships. The variable q_{imv} is governing the load or discharge from a given ship, while simultaneously being related to a specific harbour. This is handle by adding coupling constraints.

$$t^S_{im} - \sum_{v \in V} t_{imv} = 0 \tag{62}$$

$$q_{im} - \sum_{v \in V} q_{imv} = 0 \tag{63}$$

Thus, the constraints can be modified and separated into ship routing constraints, harbour inventory constraints and common constraints.

6.2.1 Ship Routing Constraints

$$W_{imv} + \sum_{j \in H_v} \sum_{n \in M_{jv}} x_{jnimv} - \sum_{j \in H_v} \sum_{n \in M_{jv}} x_{imjnv} - z_{imv} = 0, \forall v \in V, i \in H_v, m \in M_{iv} \tag{64}$$

$$x_{imjnv}(l_{imv} + J_j q_{jnv} - l_{jnv}) = 0, \forall v \in V, (i, m, j, n) \in A_v \tag{65}$$

Equation 65 can be linearized as shown in equation 66 and 67:

$$l_{imv} + J_j q_{jnv} - l_{jnv} + C^{AP}_v x_{imjnv} \leq C^{AP}_v, \forall v \in V, (i, m, j, n) \in A_v \tag{66}$$

$$l_{imv} + J_j q_{jnv} - l_{jnv} - C^{AP}_v x_{imjnv} \geq -C^{AP}_v, \forall v \in V, (i, m, j, n) \in A_v \tag{67}$$

$$l_{imv} - \sum_{j \in H_v} \sum_{n \in M_{jv}} C^{AP}_v x_{jnimv} \leq 0, \forall v \in V, i \in H_v, m \in M_{iv} \tag{68}$$

$$q_{imv} - \sum_{j \in H_v} \sum_{n \in M_{jv}} Q^{MAX}_{imv} x_{jnimv} \leq W_{imv} Q^{MAX}_{imv}, \forall v \in V, i \in H_v, m \in M_{iv} \tag{69}$$

In order to keep the variables in the subproblems separated, the y_{im} variable in **?** is replaced by a x_{jnimv} variable with the opposite sign, reflected the definition of y_{im} being 1 when no ships arrive at the given node, and 0 otherwise.

$$\sum_{v \in V} q_{imv} - \sum_{v \in V} \sum_{j \in H_v} \sum_{n \in M_{jv}} Q^{MIN}_{imv} x_{jnimv} \geq 0, \forall i \in H_T, m \in M_{Ti} \tag{70}$$

$$x_{imjnv}(t^E_{im} + T^S_{ijv} - t_{jn}) \leq 0, \forall v \in Vm(i, m, j, n) \in A_v \tag{71}$$

Equation 71 can be linearized as shown in equation 72

$$(t^E_{im} + T^S_{ijv} - t_{jn}) + M x_{imjnv} \leq M, \forall v \in Vm(i, m, j, n) \in A_v \tag{72}$$

In addition we transform constraint into a ship dependent constraint for time windows

$$\sum_{j\in H_T}\sum_{n\in M_{Ti}} T^{WS}_{im} x_{jnimv} \leq t^S_{imv} \leq \sum_{j\in H_T}\sum_{n\in M_{Ti}} T^{WE}_{im} x_{jnimv}, \forall v \in V, i \in H_T, m \in M_{Ti} \quad (73)$$

6.2.2 Harbour Inventory Constraints

The first two constraints are duplicates, appearing in both the ship routing constraints and the harbour investory constraints:

$$q_{im} - Q^{MAX}_{im} y_{im} \leq \sum_{v\in V} W_{imv} Q^{MAX}_{imv}, \forall i \in H_v, m \in M_{iv} \quad (74)$$

$$q_{im} + Q^{MIN}_{im} y_{im} \geq Q^{MIN}_{im}, \forall i \in H_T, m \in M_{Ti} \quad (75)$$

$$y_{im} - y_{i(m-1)} \geq 0, \forall i \in H_T, m \in M_{Ti} \quad (76)$$

In constraint 77 the q_{imv} variable is substituted with q_{im}.

$$t^S_{im} + T^Q_i q_{im} - t^E_{im} = 0, \forall i \in H_T, m \in M_{Ti} \quad (77)$$

$$t^S_{im} - t_{Ei(m-1)} + T^B_i y_{im} \geq T^B_i, \forall i \in H_T, m \in M_{Ti} \quad (78)$$

$$T^{WS}_{im} \leq t^S_{im} \leq T^{WE}_{im}, \forall i \in H_T, m \in M_{Ti} \quad (79)$$

In constraint 80 the q_{imv} variable is substituted with q_{im}.

$$s^S_{im} - J_i q_{im} + R_i t^E_{im} - R_i t^S_{im} - s^E_{im} = 0, \forall i \in H_T, m \in M_{Ti} \quad (80)$$

$$s_{Ei(m-1)} + R_i t^S_{im} - R_i t_{Ei(m-1)} - s^S_{im} = 0, \forall i \in H_T, m \in M_{Ti} \quad (81)$$

$$S^{MIN}_i u_i \leq s^S_{im} \leq S^{MAX}_i, \forall i \in H_T, m \in M_{Ti} \quad (82)$$

$$S^{MIN}_i u_i \leq s^E_{im} \leq S^{MAX}_i, \forall i \in H_T, m \in M_{Ti} \quad (83)$$

$$S^{EMIN}_i u_i \leq s^E_{im} + R_i(T u_i - t^E_{im}) \leq S^{MAX}_i, \forall i \in H_T, m \in |M_{Ti}| \quad (84)$$

$$S^S_i + R_i(t^E_{i1}) = s^S_{i1}, \forall i \in H_T \quad (85)$$

6.2.3 Common Constraints

$$\sum_{v\in V}\sum_{j\in H_v}\sum_{n\in M_{jv}} x_{jnimv} + y_{im} + \sum_{v\in V} W_{imv} = 1, \forall i \in H_T, m \in M_{Ti} \tag{86}$$

$$t^S_{im} - \sum_{v\in V} t_{imv} = 0 \tag{87}$$

$$q_{im} - \sum_{v\in V} q_{imv} = 0 \tag{88}$$

6.3 Dantzig-Wolfe Decomposition

By applying a Dantzig-Wolfe decomposition approach, the common constraints establish the master problem. The ship routing constraints does not include interactions between the ships, an the harbour inventory constraints does not include interactions between the harbours. Thus the sub-problems for each ship and each harbour can be solved independently. For each ship routing problem, a feasible route have to be defined. By defining X_{imjnvr} as the amount of feed carried by ship v on route r between node (i,m) and node (j,n), Q_{imvr} as the feed loaded or discharged by ship v on route r at node (i,m), and T_{imvr} as the start time for loading or discharge by ship v on route r at node (i,m), the ship routes are defined.

Similarly to the routes r for the ship routing problem, visiting sequences s are defined for the harbours. The sequences are defined by the variables T_{ims}, Q_{ims} and Y_{ims}. T_{ims} is defined as the starting time of feed loading or discharge at node (i,m) in sequence s. Q_{ims} is defined as the amount of feed loaded or discharged at node (i,m) in sequence s. Y_{ims} is equal to 1 if node (i,m) is not visited on sequence s and 0 otherwise.

By defining $S_v^{\lambda^1}$ as the feasible corner point solutions for the sub-problem for ship v, all solutions for x_{imjnv}, t_{imv} and q_{imv} satisfying the ship routing constraints, can be expressed as convex combinations of $S_v^{\lambda^1}$. x_{imjnv} must remain binary. Thus, the the following relations is established.

$$x_{imjvn} = \sum_{r\in S_v^{\lambda^1}} X_{imjnvr}\lambda^1_{vr}, \forall v \in V, (i,m,j,n) \in A_v \tag{89}$$

$$x_{imjvn} \in 0, 1, \forall v \in V, (i,m,j,n) \in A_v \tag{90}$$

$$q_{imv} = \sum_{r\in S_v^{\lambda^1}} Q_{imvr}\lambda^1_{vr}, \forall v \in V, i \in H_v, m \in M_{iv} \tag{91}$$

$$t_{imv} = \sum_{r\in S_v^{\lambda^1}} T_{imvr}\lambda^1_{vr}, \forall v \in V, i \in H_v, m \in M_{iv} \tag{92}$$

$$\sum_{r \in S_v^{\lambda^1}} \lambda_{vr}^1 = 1, \forall v \in V \tag{93}$$

$$\lambda_{vr}^1 \geq 0, \forall v \in V \tag{94}$$

The number of harbour arrivals at node (i,m) on route r by ship v can be defined as B_{imvr}. If $B_{imvr} = \sum_{j \in H_v} \sum_{n \in M_{jv}} X_{jnimvr}$, routes where $B_{imvr} \in \{0,1\}$ are considered feasible.

By defining $S_i^{\lambda^2}$ as the feasible corner point solutions for the sub-problem for harbour i, all solutions for y_{im}, t_{im} and q_{im} satisfying the ship routing constraints, can be expressed as convex combinations of $S_i^{\lambda^2}$. y_{im} must remain binary. Thus, the the following relations is established.

$$y_{im} = \sum_{s \in S_i^{\lambda^2}} Y_{ims} \lambda_{is}^2, \forall i \in H_T, m \in M_{Ti} \tag{95}$$

$$y_{im} \in 0, 1, \forall i \in H_T, m \in M_{Ti} \tag{96}$$

$$q_{im} = \sum_{s \in S_i^{\lambda^2}} Q_{ims} \lambda_{is}^2, \forall i \in H_T, m \in M_{Ti} \tag{97}$$

$$t_{im} = \sum_{s \in S_i^{\lambda^2}} T_{ims} \lambda_{is}^2, \forall i \in H_T, m \in M_{Ti} \tag{98}$$

$$\sum_{s \in S_i^{\lambda^2}} \lambda_{is}^2 = 1, \forall i \in H_T \tag{99}$$

$$\lambda_{is}^2 \geq 0, \forall i \in H_T, s \in S_i^{\lambda^2} \tag{100}$$

By substituting the the relations from equation 89-100 in the common constraints defined, the master problem is transformed into a problem with decision variables λ_{vr}^1 and λ_{is}^2, as shown below.

$$min z = \sum_{v \in V} \sum_{r \in S_v^{\lambda^1}} C_{Tvr} \lambda_{vr}^1 \tag{101}$$

$$\sum_{v \in V} \sum_{r \in S_v^{\lambda^1}} B_{imvr} \lambda_{vr}^1 + \sum_{s \in S_i^{\lambda^2}} Y_{ims} \lambda_{is}^2 + \sum_{v \in V} W_{imv} = 1, \forall i \in H_T, m \in M_{Ti} \tag{102}$$

$$\sum_{v \in V} \sum_{r \in S_v^{\lambda^1}} Q_{imvr} \lambda_{vr}^1 - \sum_{s \in S_i^{\lambda^2}} Q_{ims} \lambda_{is}^2 = 0, \forall i \in H_T, m \in M_{Ti} \tag{103}$$

$$\sum_{v \in V} \sum_{r \in S_v^{\lambda^1}} T_{imvr} \lambda_{vr}^1 - \sum_{s \in S_i^{\lambda^2}} T_{ims} \lambda_{is}^2 = 0, \forall i \in H_T, m \in M_{Ti} \tag{104}$$

$$\sum_{r \in S_v^{\lambda^1}} \lambda_{vr}^1 = 1, \forall v \in V \tag{105}$$

$$\sum_{r \in S_i^{\lambda^2}} \lambda_{is}^2 = 1, \forall i \in H_T \tag{106}$$

$$\lambda_{vr}^1 \geq 0, \forall v \in V, r \in S_v^{\lambda^1} \tag{107}$$

$$\lambda_{is}^2 \geq 0, \forall i \in H_T, s \in S_i^{\lambda^2} \tag{108}$$

$$\sum_{r \in S_v^{\lambda^1}} X_{imjnvr} \lambda_{vr}^1 \in \{0, 1\}, \forall v \in V, (i, m, j, n) \in A_v \tag{109}$$

6.4 Column generation

By generating all the feasible solutions for ship routes and harbour visiting sequences, the master problem could be solved to optimality in order to find the optimal solution for the problem. However, the number of extreme points and rays is usually growing exponentially with an increasing number of constraints and variables. Since the we are restricted by computer memory it would be impossible to store all feasible ship routes and harbour visiting sequences for larger problems. It would also take an extremely long time even to generate all feasible routes. In order to these restrictions, a decomposition algorithm with delayed column generation can be applied. The delayed column generation ensures that columns are generated only after being proven to have a negative reduced cost and is about to enter the basis. The subproblems can therefore be treated as search methods in order to find the columns of reduced negative cost.

When the subproblems are solved and have found columns with negative reduced cost, the columns are added to the master problem, which is then resolved in order to generate new dual variables. The dual variables are then used in order to generate a new objectives for the subproblems, such that new columns of reduced negative cost can be found. This procedure continues until all no columns of negative reduced cost exists.

The dual variables from the master problem have to be defined for every constraint. The dual variables D_{im}^V, D_{im}^Q, D_{im}^T, $D_v^{\lambda^1}$, $D_i^{\lambda^2}$ are defined for constraint (102)-(106). Thus, the reduced cost for the ship routing subproblem can be formulated as

$$C_{vr}^{SR} = C_{vr} - \sum_{i \in H_T} \sum_{m \in M_{Ti}} B_{imvr} D_{im}^V - \sum_{i \in H_T} \sum_{m \in M_{Ti}} Q_{imvr} D_{im}^Q - \sum_{i \in H_T} \sum_{m \in M_{Ti}} T_{imvr} D_{im}^T - D_v^{\lambda^1} \tag{110}$$

while the reduced cost for the harbour visiting sequence subproblem can be formulated as

$$C_{is}^{HV} = C_{is} - \sum_{m \in M_{Ti}} Y_{ims} D_{im}^{V} - \sum_{m \in M_{Ti}} Q_{ims} D_{im}^{Q} - \sum_{m \in M_{Ti}} T_{ims} D_{im}^{T} - D_{i}^{\lambda^2} \tag{111}$$

6.5 Subproblems

The subproblems are solved in order to generate columns of negative reduced cost. Both the ship routing subproblem and the harbour visiting sequence subproblem can be defined as shortest path problems solved by dynamic programming algorithms.

An easier structure of the subproblems can be obtained by discretizing load quantities. In the ship routing subproblem different nodes can be formulated for each predefined load quantity, while in the harbour visiting sequence subproblem nodes are generated for the cumulative load quantity.

6.5.1 Ship routing subproblem

The ship routing subproblems is solved in order to find the route of least cost. Constraints concerning routing are placed in this subproblem. In addition the loading and discharge quantities have to be considered, as the ships are subject to capacity constraints. As no constraints covers more than one ship, the subproblems for each ship can be solved independently.

$$\begin{aligned} min \sum_{i \in H_T} \sum_{m \in M_{Ti}} \sum_{j \in H_T} \sum_{n \in M_{Ti}} C_{ijv} x_{imjnv} - \sum_{i \in H_T} \sum_{m \in M_{Ti}} \sum_{j \in H_T} \sum_{n \in M_{Ti}} x_{imjnv} D_{im}^{V} \\ - \sum_{i \in H_T} \sum_{m \in M_{Ti}} q_{imv} D_{im}^{Q} - \sum_{i \in H_T} \sum_{m \in M_{Ti}} t_{imv} D_{im}^{T} \end{aligned} \tag{112}$$

It can be be seen that equation (112) does not include the term $D_v^{\lambda^1}$ found in equation (110). $D_v^{\lambda^1}$ is a constant in the subproblem and will not influence the optimal solution. Thus, a given ship route is only implemented in the master problem if the objective is less than $D_v^{\lambda^1}$.

6.5.2 Harbour visiting sequence subproblem

Since no constraints cover more than one port, the harbour visiting sequence subproblems can be solved independently for all harbours. Load and time information is not given in advance when solving the subproblems, while the dual variables from the master problem remains constant. The objective function of harbour visiting sequence subproblem can thus be formulated

$$min\ -\sum_{m\in M_{Ti}} y_{im}D^{V}_{im} - \sum_{m\in M_{Ti}} q_{im}D^{Q}_{im} - \sum_{m\in M_{Ti}} t_{im}D^{T}_{im} \tag{113}$$

It can be be seen that equation (113) does not include the term $D_i^{\lambda^2}$ found in equation (111). $D_i^{\lambda^2}$ is a constant in the subproblem and will not influence the optimal solution. Thus, a given sequence is only implemented in the master problem if the objective is less than $D_i^{\lambda^2}$. It can also be seen that there is no real cost to the transportation problem. Only the cost associated with the dual variables make up the reduced cost, and thus the subproblem objective.

6.6 Discussion

The Dantzig-Wolfe decomposition is widely used in large scale optimization problems. A major advantage with the Dantzig-Wolfe decomposition approach is that the columns are generated and added to the master problem only when proven to have a negative reduced cost. Thus, problems of large size can be solved without being limited by memory restriction to the same extent as a simpler simplex based method would. The efficiency of the Dantzig-Wolfe decomposition in terms of computational time is debated in litterateur. While Bertsimas and Tsitsiklis (1997) claims there are usually only minor improvements in computational time as compared to a simplex based algorithms,Christiansen (1999) claims to have significantly reduced computational time using this approach for an IRP. Improved solution time would enhance the applicability of a optimization tool for fish farmers, as decisions often have to be settled rapidly. Savings in computational time is also essential in order to minimize costs of running the model.

While Christiansen (1999) introduces the coupling constraints and the model for the master problem used in the Dantzig-Wolfe decomposition proposed here, additional work have been done by stating the objectives for the subproblems and the use of dual variables from the master problem in said objectives. In addition the option to select starting nodes are added. As no branch-and-bound system based on the underlying variables are implemented, the Dantzig-Wolfe decomposition is not tested in the computational study.

7 Clustering of nodes

As the IRP model represents a NP-hard problem, the complexity of the problem increases exponentially as the number of nodes increase. Modelling all the 150 fish farming locations that Mowi have along the Norwegian coast would lead to a problem formulation that is practically impossible to solve to the global optimum. Even finding feasible solutions would be resource intensive, and thus it would also be difficult to find upper bounds on the minimization problem. In order to limit the number of variables in the problem, clustering of nodes is applied. This approach is realistic, as ships often visits nodes that are close to each other when sailing. By clustering nodes, one limits the number of variables in the system, making it easier to find good bounds for the problem. Better bounds limits the integrality gap. However, in order to obtain realistic results when modelling, careful considerations have to be taken into account when determining the the size and composition of clusters.

7.1 Clustering methods

Theoretical or manual clustering methods can be used in order to generate suitable clusters. Haugland and Thygesen (2017) provides a overview of the of most used clustering methods and a review of their key findings is following.

7.1.1 Theoretical clustering

MacQueen et al. (1967) formally introduced the k-means method for clustering of nodes, but the idea was first presented in Steinhaus (1956). This clustering method is one of the most popular in clustering and also one of the most used methods in data mining today. The aim of the method is to partition n nodes into k clusters, such that each node is partitioned into the clusters with the nearest mean.

While similar to the k-means method in the sense that n nodes are clustered into k clusters, the k-median differs in the way that each node is assigned to the closest median of a given cluster instead of the mean. This makes the method more resilient to outlier nodes that can shift the the center of the cluster substantially (Whelan et al., 2015).

Hierarchical clustering is a method that work by adding the node that is closest to a cluster into the given cluster. The method starts out with all nodes defined a cluster and runs until the desired number of clusters is reached or until there is only one cluster left.

7.1.2 Manual clustering

As the Norwegian coast line is uneven, with many fjords and island, the sailing distance between locations may not be equal to the most direct path between the points. Such uneven conditions make theoretical approaches less suitable. Thus manual clustering may be seen as a useful alternative. Manual clustering approaches can be carried out by simply looking at the map and selecting clusters based on wanted criteria.

One way of manually clustering the nodes is to divide the Norwegian coast into regions, divided from south to north. As the Norwegian coast may be seen as a continuous line from south to north, the regions can simply be split by drawing latitudinal lines limiting the regions. Alternatively, Mowi's own region divisions can be used as shown in the figure below. However, such a division leads to only 3 regions, which is not sufficient in order to model the complexity and differences of the fish farms included in the regions.

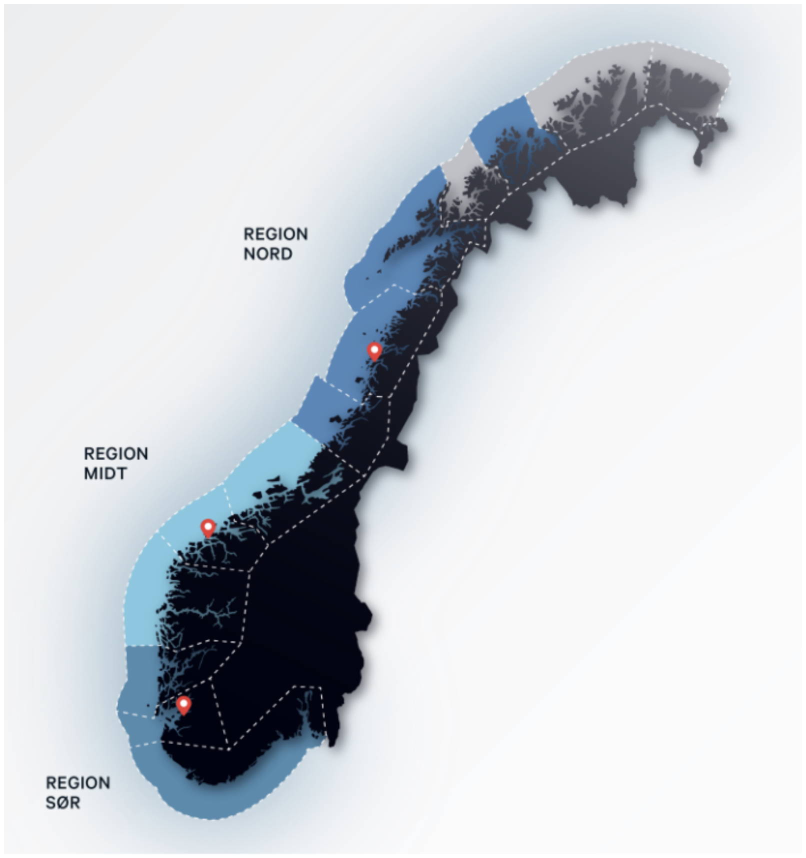

Figure 9: Norwegian Regions for Mowi. Source: Marine Harvest (2018b)

The Government of Norway have decided to make so called production zones along the coast to reduce the risk of disease spreading and genetic impact on wild fish from escaped farmed fish (Lekve, 2016). The Institution of Ocean Research and The Norwegian Directorate of Fisheries was asked to come up with a proposal for how this could be done. In Ådlandsvik (2015), it was proposed to divide the Norwegian coast into 11 to 13 production zones.
In the report, a influence matrix that showed the potential contamination between fish farming locations were used in order to generate the production zones.

A third manual clustering method is to group farms that are close to each other. As fish

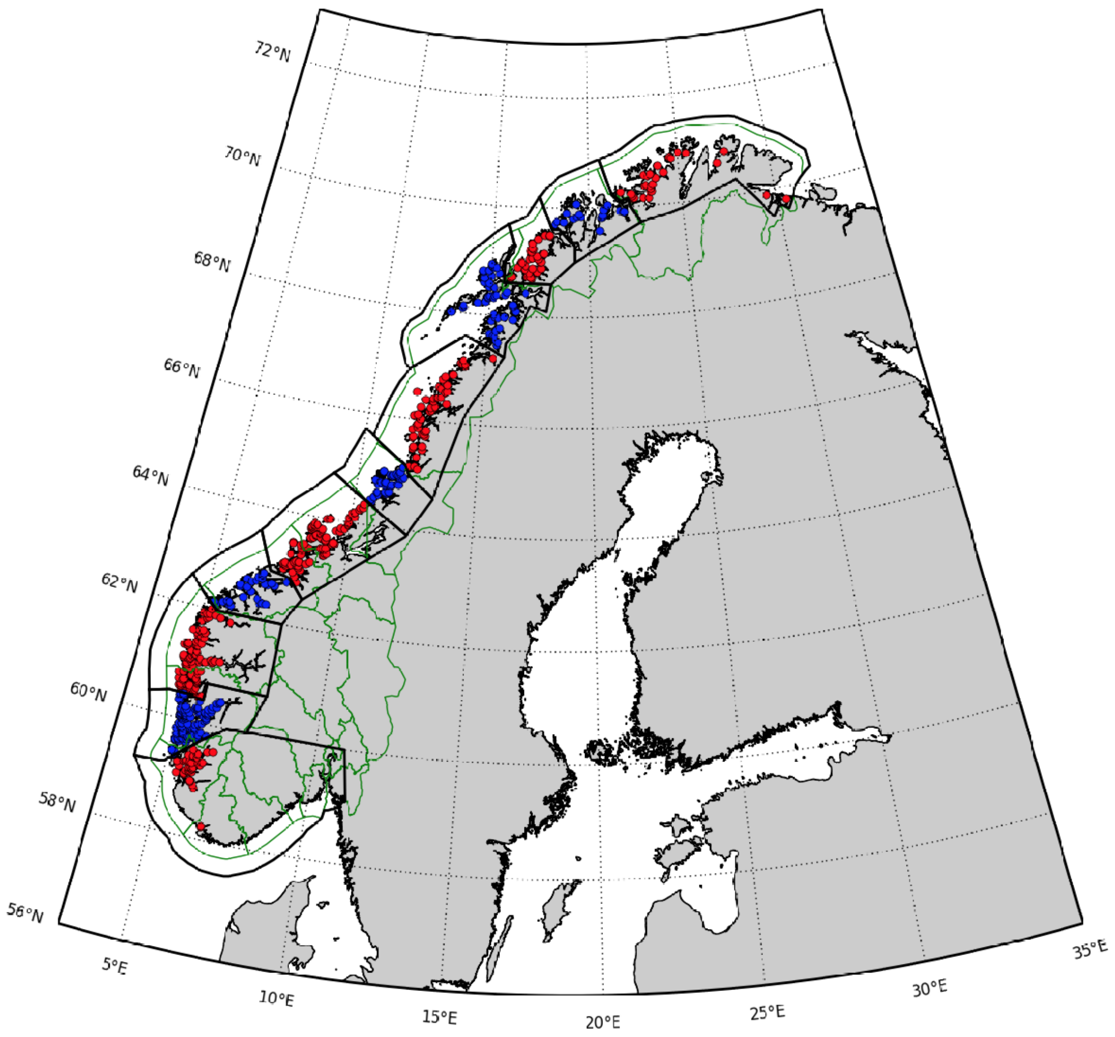

Figure 10: Production zones. Black lines divide the production zones. Green lines separate the counties. Nodes in every other zone are red and blue respectively. Source: Ådlandsvik (2015)

farming is mostly done in shielded locations less subject to hash weather, the fish farms for given companies are often naturally clustered into smaller areas as fjords or islands shielding the locations. By using these clusters, one is able to model the problem in a realistic way, as the clusters are often subject to the same temperature and weather, thus keeping the consumption rates fairly similar among the locations in the clusters. The downside to using this method, is that it is hard to model with computer algorithms and have to be modelled manually, witch can be time consuming.

7.2 Implementation of clustering methods

7.2.1 Production zones

By using the production zones defined in Ådlandsvik (2015), the clusters can be made by assigning the given fish farms operated by Mowi in the area to the respective cluster. The center of each cluster can then be calculated as the average of the locations for the nodes in the cluster. While the implementation of these production zones may limit what ship can sail within each cluster, the importance of having defined these clusters gain substantial importance. In the future the fish farming companies may be subject to regulations forcing the companies to have depots within each sector. If these regulations are enforced, only minor changes have to be made for the IRP model already defined, as we can model the depots as consumption nodes.

7.2.2 Natural selection

As the natural methods are based on manually selection nodes in a strategic way, this process can be time consuming. By laying out a map and selecting nodes for each cluster, cluster by cluster, one is able to obtain a fairly good solutions that take fjord, islands and other obstacles into account. This operation was performed in Haugland and Thygesen (2017), and yielded the result as shown in the figure below:

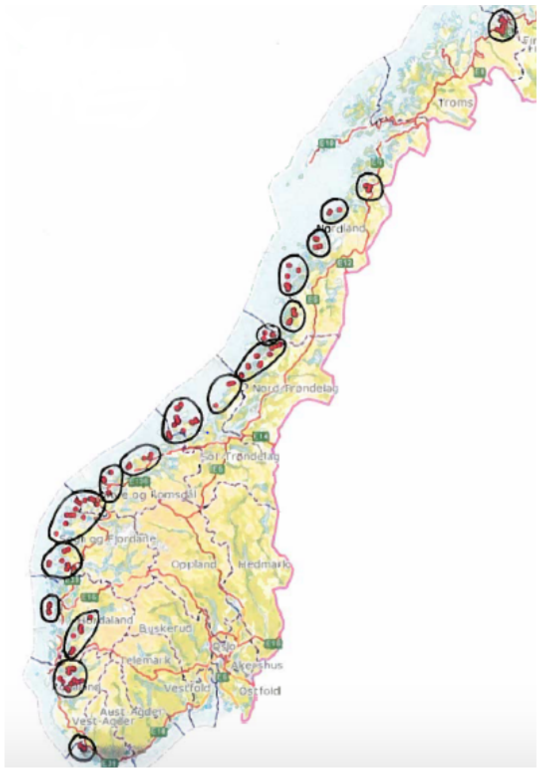

Figure 11: Natural selection of clusters. Source: Haugland and Thygesen (2017)

7.2.3 K-Means Algorithm

The simplicity of the k-means algorithm, makes it easy to implement with a computer programming language. The algorithm also works well with large data sets, allowing fast computation times, even with a high number of fish farms as input data. MATLAB has a built in function named kmeans, that performs the k-means clustering of nodes given by input data. It returns a vector containing cluster indices of each node, the location of the cluster centers and the distance from each node to its respective cluster center.

7.3 Estimation of Parameters

When nodes have been clustered, a set of parameters have to be estimated for the new system of clusters. The two most important parameters are the distance between clusters and the internal sailing distance in the clusters.

7.3.1 Estimation of Distance between Clusters

The fish farms are located at various locations along the Norwegian cost-line. Many of the fish farms are located in fjords, and islands and shallow waters restricts vessels from sailing in straight lines when travelling between fish farms. More sophisticated methods therefore have to be applied when estimating travelling distances between clusters. Also where to set the location of a cluster is a challenge, as there are multiple nodes of different locations inside each cluster.

There are multiple ways to set the location of cluster. One way is to take the average of all node locations. This can either be done giving all nodes an equal weighting or the average may include different weights depending on the consumption at each node. These can both include new challenges, as the average may be on land or other areas inaccessible to the vessel. Another approach is to make the first farm that is visited in a cluster a supernode, such that the location of the cluster as a whole is set to the same location as the supernode.

Regardless of the approach is used to set the location of the cluster, the main problem is still to find the sailing distance between clusters. The data used for calculations in this thesis is based on sailing distances generated by Searoutes.com. Searoutes allows the user to manually select start and ending positions for ships with a user-friendly interface and generates feasible routes for sailing. However, it should be noted that these estimates are not necessarily 100% accurate, and ship captains with knowledge about the local sailing areas should be consulted when generating actual route estimates.

7.3.2 Estimation of Internal Sailing Distances in Clusters

As nodes inside a cluster are not necessarily in close proximity to each other, sailing between the nodes internally in a cluster have to be accounted for. The internal sailing can be modelled as travelling salesmen problems, with only the nodes in the cluster considered. While this would yield accurate results, it is also very time consuming. As multiple test cases are considered in this thesis, a simpler method is considered. Coordinates for cluster centers are generated and coordinates for nodes are known. Thus, the distance between cluster centers and individual nodes can be calculated. For each cluster the distances between cluster centers and individual nodes are summarized. As the summarized traveling distances is a clear overestimate relative to the actual traveling distance, the summarized distances between cluster centers and individual nodes are divided by 2.

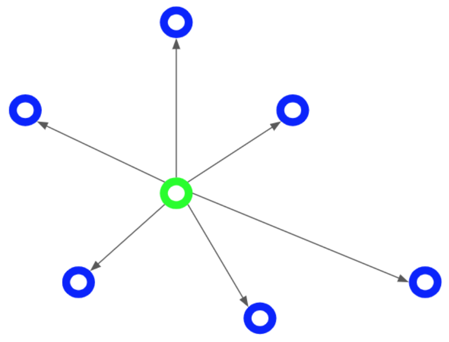

Figure 12: Estimate for internal sailing distance

7.4 Inventory Restrictions for Clustered Customers

When aggregating the customer nodes, a number of problems have to be handled. The first one is the consumption rate. The consumption rate of the cluster can be modelled as the sum of the consumption at all the nodes in the cluster. A more complex problem to handle is the inventory levels and the timing of the feed discharge. While also the inventory levels for a cluster can be modelled as the sum of inventory for the customers in the cluster, this approach does not take into account that a given customer in the cluster can run out of inventory before the cluster as a whole, the sum of inventory for all customers in the cluster, runs out of inventory. This can be handled by different strategies with modifications to the model and by setting a number of assumptions.

One strategy to ensure that inventory restrictions are satisfied for all customers when clustered, is to assume a equal consumption adjusted feed discharge for all customers in a cluster. We define i as single customers and G^k as the set of customers in cluster k. Thus, formally we have that

$$\frac{q_{inv}}{R_i} = \frac{q_{jnv}}{R_j}, \forall i, j \subseteq G^k, v \in V, n \in M_{Ti}. \tag{114}$$

If it is further assumed that the consumption adjusted capacity and initial inventory for all customers in a cluster is equal, formally

$$\frac{S_i^S}{R_i} = \frac{S_j^S}{R_j}, \forall i, j \in G^k, v \in V, n \in M_{Ti} \tag{115}$$

and

$$\frac{S_i^{MAX}}{R_i} = \frac{S_j^{MAX}}{R_j}, \forall i, j \in G^k, v \in V, n \in M_{Ti} \tag{116}$$

then it can ensured that no single customer runs out of inventory, by setting a minimum inventory level for the cluster as a whole. If A_v^k is a set of arcs that are sailed internally in cluster k by vessel v such that each customer is visited exactly once, e represents the last visited customer in the set G^k and A_v^e is the arc out from the last customer, the minimum inventory level for the cluster have to be set according to equation (117) when the first discharge operation starts, in order to ensure that no customers run out of inventory.

$$S_k^{MIN} \geq (\sum_{i \in G^k} R_i) \cdot (\sum_{i \in G^k - e} T_i^Q q_{inv} + \sum_{i,j \in A_v^k - A_v^e} T_{ijv}^S) \tag{117}$$

With this additional constraint for minimum inventory, constraint 17 from the original problem formulation is still sufficient to ensure that no customers are running out of inventory, even for an aggregated version of the problem.

7.5 Dis-Aggregation and Post-Processing

If the visiting orders internally in clusters are known, a feasible solution with each single fish farm can be generated. With such a solution available, this can be used as a starting point for further optimization. The initial solution can be implemented as hints in solvers. The commercial solver Gurobi added the possibility to use variable hints in version 6.5. The hints affects the heuristics that the commercial solver applies when solving an optimization problem (Gurobi Optimization, 2018). For mixed-integer problems the commercial solver also uses the hints for the branching decisions when exploring the search tree. High quality hints will lead to high quality solutions being found faster. While hints of low quality will lead to wasted some efforts, performance will likely not be severely impacted, as the algorithms goes on to explore different solutions. MIP Starts can also be applied in commercial solvers as Gurobi, but differs from variable hints in the way that they try to generate a single feasible solution, as opposed to the hints that impact the entire solution process. While these strategies will not be tested in this thesis, it is an interesting field for further studies.

8 Implementation and Solution Methods

This section will be focused on possible ways to implement and solve the IRP that is formulated. A method for implementing the problem on matrix form and possible solution methods are discussed.

8.1 Rewriting Problem on Matrix Form

The mathematical formulation given in section 5.2 needs to be implemented in a formal programming language in order to solve the problem. The arc-load problem is rewritten on the matrix form

$$min\, c^T x \tag{118}$$

such that

$$Ax \leq b \tag{119}$$

$$A^{eq}x = b^{eq}, \tag{120}$$

with a script made in MATLAB. The script can be found in Appendix A5. A similar script for the arc-flow model can be found in Appendix A6. Here the variables are defined as x, and when multiplied with c, this represents the objective function. Matrices A and A^{eq} represents all the variable multipliers in the constraints in the model formulation. Each row in the matrices represents one constraint and each column represents the constants multiplied with the variable of the same number. The vectors b and b^{eq} represents the constant in each constraint. Thus the number of columns in the A and A^{eq} matrices is equal to the number of variables.

By implementing the script on matrix form, it is easy to apply different computer algorithms to solve the problem. In addition, the problem can be solved using commercial solvers, as shown in section 8.3.

The implementation of the matrix form is done by first assigning a number for each variable. The total number of variables is dependent on the number of ships, clusters, possible arrivals at each cluster and the possible arcs that are feasible. In the matrix implementation, travel between all nodes have been considered feasible, thus yielding a number of variables that are only dependent on number of ships, clusters, possible arrivals at each cluster.

When all variables are assigned a given index number, the script continues by generating the vector c for the objective function, before all constraints are implemented by assigning values to the matrix A and the vector b. Also an equality or inequality characterization is assigned for each constraint. At the end the variables are assigned a variable type characterization, set to either binary, continuous or integer.

8.2 Branch-and-bound

A general branch-and-bound algorithm for solving mixed integer problems with binary variables was implemented in MATLAB, as seen in Appendix A7. The model input is the same model as generated for Gurobi solver, as shown in Appendix A5. The algorithm works by linear relaxation of the problem, where binary variables are replaced with continuous variables and new constraints limiting the feasible range of the variable from 0 to 1. The Gurobi solver is then used solely for solving the linear relaxation, and branches are made by splitting the feasible region based on the first variable number in the solution that fails to comply with the binary constraint. For each iteration a branch is checked against the upper and lower level bounds found for the problem. Infeasible solutions and solution of linear relaxations lower than upper bounds found, are discarded. The algorithm can run for a given number of iterations, time or until the upper bound is equal to the lower bound. If the upper bound is equal to the lower bound, the optimal solution is found.

The algorithm applies a breadth-first search, that explores all non-discarded nodes at a given depth before moving on to the next depth in the tree. Other implementations may be considered better, as a depth-first approach will be better for problems were heuristics are not used to provide initial solutions. The depth-first algorithm will usually be able to find a feasible solution quicker, and therefore also obtain a upper bound on the minimization problem quicker (Mehlhorn and Sanders, 2008).

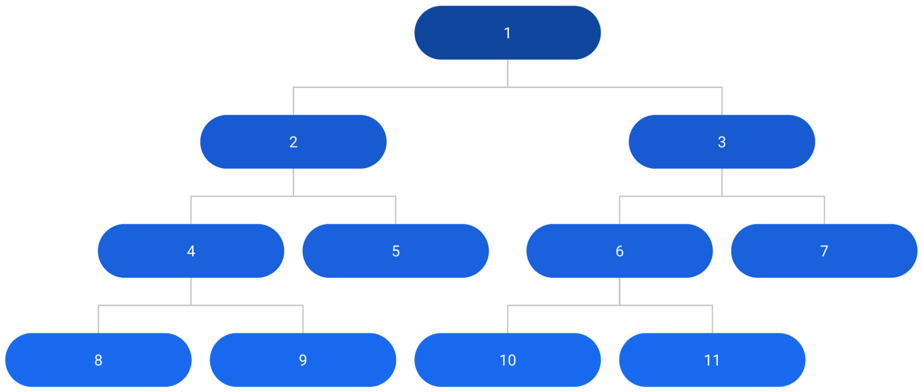

Figure 13: Breadth first search visualized. Numbers in chart represents the order of which the calculations are performed.

As the computational performance of the simple branch-and-bound is not comparable to commercial solvers, the simple branch-and-bound algorithm is not included in the computational study.

8.3 Commercial Solvers

There exists hundreds of optimization solvers, and both free and open source, as well as proprietary alternatives are available. Some solvers are more focused on certain types of optimization problems. Some of the more known solvers used for integrer problems includes FICO Xpress, Gurobi and CPLEX. As Gurobi offers easy access to academic licences with an unrestricted number of constraints and variables, this is used in order to solve the feed distribution problem. Gurobi is founded by former developers of the ILOG group that developed significant parts of the CPLEX solver.

By inputting the matrices generated in MATLAB into Gurobi Optimizer 8.1, the problems are solved. As Gurobi applies advanced heuristics, cutting plane-algorithms and branch-and-bound in order to solve the optimization problem, the problem is solved in an effective manner, making it very demanding to come up with more efficient algorithms. Gurobi is used for all test cases in the computational study.

9 Computational Study

The computational study includes results of computational time and integrality gaps for combinations of test cases, model formulations and tightening constraints. Sections for testing compatibility scenarios, consumption scenarios and robustness are also included.

9.1 Hardware

The algorithms used to solve the problem formulations have been run on a computer with a 2.6GHz dual-core Intel Core i5 processor (Turbo Boost up to 3.1GHz), and 8GB of 1600MHz DDR3L memory.

9.2 Test instances

The test instances are named accord to the test case, model formulations and tightening constraints used for the given test instances. The test case is made up of the number of nodes and clustering method used in order to generate the test case.

Table 2: Abbreviations for test cases

Test case size	Abbreviation
12 harbours	12
19 harbours	19
24 harbours	24
Clustering method	
Natural	N
Kmeans	K
Production zones	Z
Formulation	
Arc-Load	AL
Arc-Flow	AF
Formulation Improvements	
Standard Formulation	S
Time windows	TW
Sub-tour elimination	ST
Sub-tour elimination Aggregated	STA
Clique inequalities	CI

9.3 System Dimensions

The system dimensions used are based on data from Haugland and Thygesen (2017). It should be noted that this data may be outdated. However, the input data gives a realistic view of aquaculture operations and the parameters for systems included in the operations.

Mowi have a total of 4 vessels available, in addition to externally hired vessels. The 4 vessels are operated by Egil Ulvan Rederi, but is referred to as self-operated vessel later in the computational study, in order to distinguish them from externally hired vessels. Two of the vessels are LNG powered and have a cargo capacity of 3000 tonnes each. The other two vessels are are running on an IFO-type of fuel, and have cargo capacity of 1500 tonnes each. Fuel consumption for the vessels are estimated to 0.35 and 0.17 tonnes per hour for the large and small ships respectively.

Table 3: Attributes of vessels

Attributes	Vessel 1	Vessel 2	Vessel 3	Vessel 4
Fuel	LNG	LNG	IFO	IFO
Cargo capacity	3000 t	3000 t	1500 t	1500 t
Speed	13 knots	13 knots	10 knots	10 knots
Fuel consumption	0.35 t/h	0.35 t/h	0.17 t/h	0.17 t/h
Fuel price	\$260/t	\$260/t	\$310/t	\$310/t

Planning periods are set to one week. As fish farms are visited one to two times a week, problems are kept at feasible test sizes that can be solved to near optimality. It also fits well with the aim of making decision support tool for the operational stage of the planning.

Setting the number of possible arrivals correctly is a challenge when modelling the problem as an IRP. A high number of possible arrivals will increase flexibility, but also increase computational complexity. For the production zones test case a total of 5 possible arrivals are used for the feed factory. Clusters with medium consumption above 2500 tonnes a week are allowed 2 arrivals. Clusters that serves as the starting node of a vessel are allowed 2 arrivals, where one of the arrivals are the starting position itself. The vessels are allowed to discharge feed at starting node. The rest of the clusters are allowed 1 arrival. For the larger test cases, only 1 arrival is allowed at each cluster.

The staring positions of the vessels are spread out over the different regions, in order to model a realistic scenario and avoid a queue for loading feed at the feed factory at the start of the planning period. As the clusters locations differs between the different test cases, the staring positions for the vessels will not be equal for all test cases. While this can benefit costs for one test case over another, the effect should not be very large. For the test cases, there are no restrictions setting the cargo levels at the start of the planning period. Thus, the optimal solutions will naturally indicate that vessels are fully or near fully loaded at the start of the planning periods.

For fish farms having to be served by external vessels, a fixed price \$500 is added to the cost for the planning period. This cost represents the cost of hiring external vessels. In addition a cost of \$1 is added for every tonnes that is delivered to the fish farm. The \$1 cost represents the marginal cost for feed, as feed usually have to be bought from external fish feed producers when hiring external vessels. The unit cost of feed is usually higher when bought from an external company, than the production cost at the self-owned feed factory.

The assumed demand at each cluster is directly proportional to the number of fish farms in the cluster. A weekly demand of 100 tonnes of feed for every fish farm is assumed, with the medium demand case. Thus, a cluster with 2 fish farms are modelled with a demand of 200 tonnes, 3 fish farms gives 300 tonnes of demand and so on. Other consumption scenarios are tested in section 9.9. Each fish farm is modelled with storage capacities of 300 tonnes. In reality this number usually varies between 200 tonnes and 800 tonnes per fish farm.

The staring inventory levels are set to 40% of the storage capacity for the clusters. At the end of the planning horizon the inventory levels have to be equal to or higher than 40%, in order to avoid end of horizon effects. The minimum inventory levels are set to 0 during the planning period, for the test cases. Other scenarios for minimum inventory levels during the planning period are tested in section 9.10.

The travelling distances, both between clusters and internally in clusters, are given from the clustering methods. Thus travelling times are dependent on the speed of the vessels. Loading and discharge rates are set to 250 tonnes per hour. This is in line with the Mowis new vessels that have an offloading capacity of 250 tonnes per hour, and a loading capacity of 300 tonnes per hour. In order to avoid artificially small deliveries, a minimum discharge quantity of 10 tonnes is set.

9.4 Test Case - Production zones

There are a total of 11 production zones. All the fish farms in each production zone are clustered and modelled as one harbour per production zone. As the factory is an independent harbour, a total of 12 harbours are modelled.

9.4.1 Variables

The arc-flow formulation have a 1968 continuous variables before presolve, compared to only 448 continuous variables for the arc-load formulation. The difference is due the number of l_{imjnv} variables are higher than the number of l_{imv} variables that are replaced.

Table 4: Variables for production zones test case

	12Z-AL-S	12Z-AF-S
Continuous variables	448	1968
Continuous variables post presolve	184	1137
Binary variables	1712	1712
Binary variables post presove	1110	1109
Constraints	5300	3780
Constraints post presolve	3409	2320

Despite the increased number of variables, the number of constraints are much lower in arc-flow formulation. Before presolve, the number of constraints for the arc-flow formulation is 3780, as compared to 5300 for the arc-load formulation. The lower number of constraints is due to the replacement of constraint 5 found in the arc-flow formulation. Constraint 5 is implemented in the solver by linearizing the constraint into constraint 6 and 6. Both of these constraints contains a number of sub-constraints equal to the number of possible arcs, multiplied by the number of ships. The arc-flow formulation replaces these constraints by a single set of constraints that does not need to be linearized. The lack of this linearization in the model also limits the computational inefficiencies related to Big M formulations.

9.4.2 Results

All model formulations for the production zones test case are solved to optimality before the time limit of 1800 seconds is reached. As expected, all model formulations also generated the same solution.

Table 5: Results for the production zones test case

Instance	Running Time	Solution	Bound	Integrality Gap
12Z-AL-S	340s	13340	13340	0%
12Z-AL-TW	536s	13340	13340	0%
12Z-AL-ST	93s	13340	13340	0%
12Z-AL-STA	104s	13340	13340	0%
12Z-AL-CI	109s	13340	13340	0%
12Z-AF-S	30s	13340	13340	0%
12Z-AF-TW	43s	13340	13340	0%
12Z-AF-ST	67s	13340	13340	0%
12Z-AF-STA	33s	13340	13340	0%
12Z-AF-CI	46s	13340	13340	0%

While all model formulations are solved to optimality, the running time varies a lot between them. The arc-flow models are solved faster than the arc-load models for formulation improvements tested. Among the arc-load formulations, the one with sub-tour elimination constraints is solved fastest, with a running time of 93 seconds. However, even this formulation is solved slower than the slowest arc-flow formulation. The standard arc-flow formulation is solved in 30 seconds, more than 11 times faster than the standard arc-load formulation that is solved in 340 seconds. The standard arc-flow formulation is also the formulation that is solved fastest among all tested.

The tightening of time windows is done by forcing all deliveries to happen during the first 6 days of the planning period. Both the arc-flow and the arc-load formulation has the same solution being found with the tightened time windows and the standard formulations. When adding the tightened time windows, the average fleet utilization increases from 74% to 86%. However, the computational time increases in both cases.

Both the aggregated and split sub-tour elimination constraints improves the computational time of the arc-load formulation significantly for the test set with production zones. The split sub-tour elimination constraints is the most efficient formulation improvement, with a 73% reduction in computational time compared to the standard arc-load formulation. For the arc-flow formulation neither the aggregated nor the split sub-tour elimination constraints leads to improved computational times. The arc-flow formulation with split sub-tour elimination constraints leads to a 123% increase in computational time as compared to the standard arc-flow formulation.

Similar to the sub-tour aggregation constraints, the clique inequalities improves the computational time for the arc-load formulation, while it increases computational time for the arc-flow formulation. For the arc-load formulation the computational time is reduced by 68%, to 109 seconds. For the arc-flow formulation the computational time is increased by 53%, to 46 seconds.

The solution for routing of the vessels are almost identical for all formulations. Since all problems are solved to optimality with the same solution objective, the solutions only differs by arrival times and load and discharge quantities. The same arcs are used by similar sized vessels for all problem formulations. The routing shown in figure 14 have the largest vessels, plotted with yellow and gray arcs, serve the southern and northern fish farms. The small vessels mainly serve the fish farms close to the feed factory. The factory is lables as node number 1, while node 2 is the northernmost fish farm, and node 12 is the southernmost fish farms.

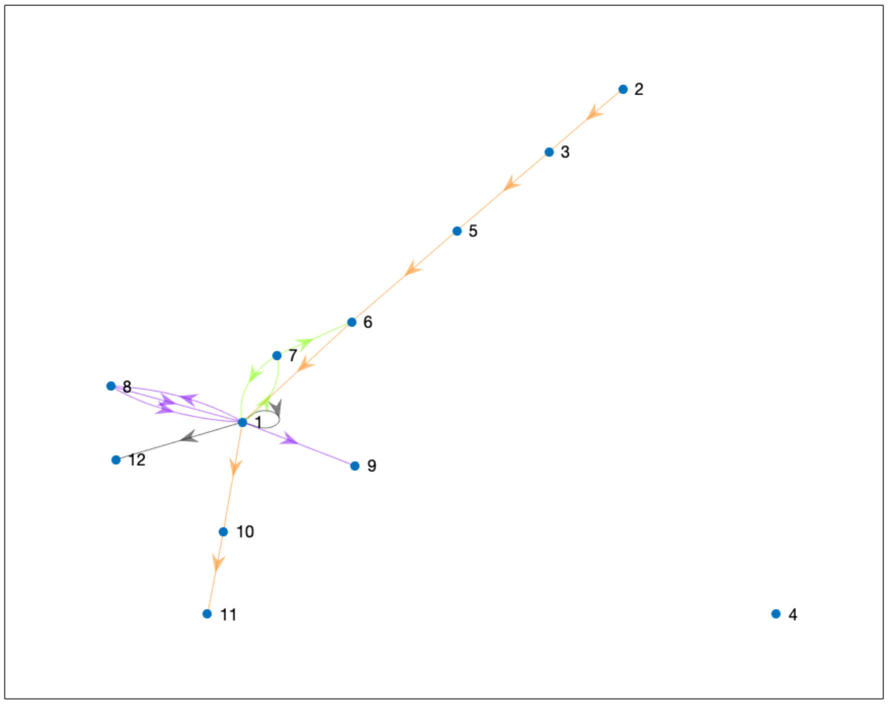

Figure 14: Routing for production zones test case. Nodes are denoted by node number. Node locations are arbitrary and does not indicate the real locations of the nodes. Ships are separated by arc colour.

9.5 Test Case - Natural Selection

With the natural selection method, a total of 18 clusters of fish farms were selected. Each cluster is modelled as one harbour per production zone. As the factory is an independent harbour, a total of 19 harbours are modelled.

9.5.1 Variables

The arc-flow formulation have a 2328 continuous variables before presolve, compared to only 480 continuous variables for the arc-load formulation. Similar to the production zones test case, the difference is due to the number of l_{imjnv} variables being higher than the number of l_{imv} variables that are replaced.

Table 6: Variables for natural selection test case

	19Z-AL-S	19Z-AF-S
Continuous variables	480	2328
Continuous variables post presolve	211	1433
Binary variables	2065	2065
Binary variables post presove	1404	1404
Constraints	6358	4510
Constraints post presolve	4291	2930

9.5.2 Results

None of the problem formulations for the natural selection test case are solved to optimality. The test instances are stopped manually after a running time of 1800 seconds. The arc-flow formulation with aggregated sub-tour elimination constraints finds the best solution objective of $11 519, and also have the lowest integrality gap with 8.7%. Combining the solution of the arc-flow formulation with aggregated sub-tour elimination constraints with the bound of the standard arc-flow formulation, a integrality gap of 7.3% is obtained.

Table 7: Results for the natural selection test case

Instance	Running Time	Solution	Bound	Integrality Gap
19N-AL-S	1800s	16900	6771	59.9%
19N-AL-TW	1800s	15258	7039	53.9%
19N-AL-ST	1800s	12935	6243	51.7%
19N-AL-STA	1800s	12125	6802	43.9%
19N-AL-CI	1800s	12132	8161	32.7%
19N-AF-S	1800s	11723	10680	8.9%
19N-AF-TW	1800s	12213	10525	13.7%
19N-AF-ST	1800s	11519	9878	14.2%
19N-AF-STA	1800s	11519	10517	8.7%
19N-AF-CI	1800s	11859	10338	12.8%

The arc-flow formulation proves superior in every aspect, consistently yielding better solutions and bounds, regardless of which formulation improvements are added. The standard arc-load formulation provides a integrality gap of 59.9%, significantly more than the integrality gap of 8.9% provided with the standard arc-flow formulation.

The arc-load formulation with tightened time windows yields a better solution and bound, than the standard arc-load formulation. The lower bound of the arc-load formulation with tightened time windows is less than the best solution found for any formulation, thus one can not conclude whether the tightened time windows affects the optimal solution objective. The arc-flow formulation with tightened time windows gives a worse solution and bound, than the standard arc-flow formulation. The arc-flow formulation with tightened time windows yields a integrality gap of 13.7% and a solution of $ 12 213.

Both the aggregated and the split sub-tour elimination constraints improves integrality gaps and solutions for the the arc-load formulation. The aggregated sub-tour elimination constraints gives the best solution and bound. This results in a integrality gap of 43.9%, compared to 59.9% for the standard formulation, both with 1800 seconds of running time. The sub-tour elimination constraints for the arc-flow formulation has a larger integrality gap than the arc-flow formulation without formulation improvements. However, this is due to the bound being lower. The solution found is better than that of the standard arc-flow formulation, and the solution is found after only about 600 seconds, as compared to a total run time of 1800 seconds. The aggregated sub-tour elimination constraints for the arc-flow formulation also has a lower integrality gap than the arc-flow formulation without formulation improvements. This is despite having a worse bound. The main improvement is therefore attributable to the lower solution found. Adding clique inequalities improves the solution of the arc-load formulation significantly, and reduces the integrality gap from 59.9% to 32.7%. The effect on the the arc-flow formulation is more limited, with a slight increase in the solution objective relative to the standard arc-flow formulation.

While the different formulations have unequal solutions, the routing solution found with the arc-flow formulation is presented in figure 15. The routing shown in the figure have the largest vessels, plotted with yellow and gray arcs, serve the southern and northern fish farms. The small vessels mainly serve the fish farms close to the feed factory. The main exception is one of the small vessels serving cluster 15 to the south. The factory is labelled as node number 1, while node 2 is the northernmost fish farm, and node 19 is the southernmost fish farms.

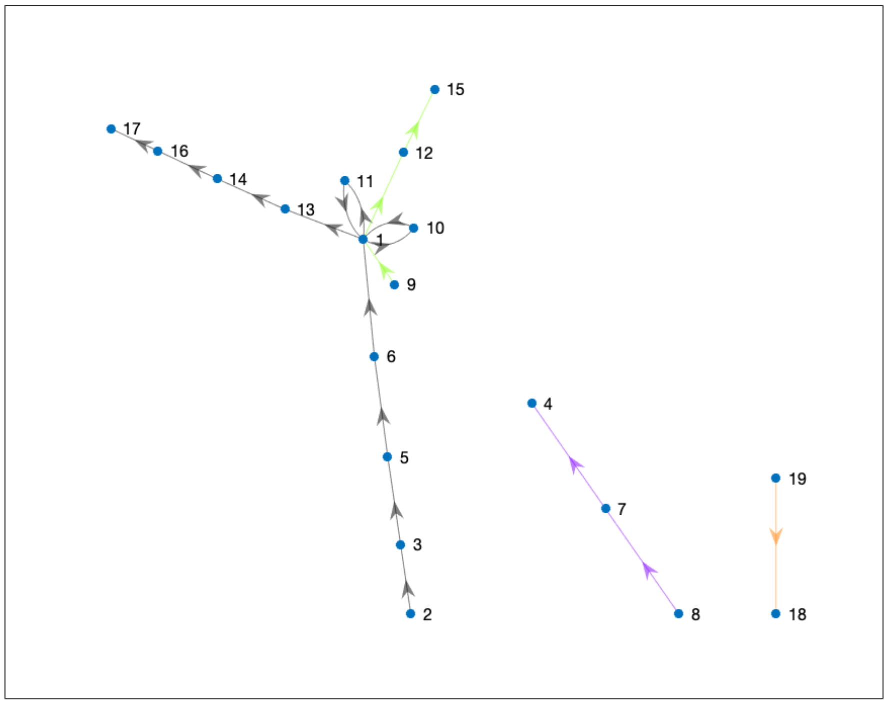

Figure 15: Routing for natural selection test case. Nodes are denoted by node number. Node locations are arbitrary and does not indicate the real locations of the nodes. Ships are separated by arc colour.

9.6 Test Case - K-Means

The K-Means test case is generated with 23 clusters of fish farms. Thus, including the feed factory, the total number of harbours that is modelled is 24.

9.6.1 Variables

Table 8: Variables for the K-Means test case

	24K-AL-S	24K-AF-S
Continuous variables	704	3728
Continuous variables post presolve	277	2545
Binary variables	3300	3300
Binary variables post presove	2447	2447
Constraints	10108	7084
Constraints post presolve	7472	5108

9.6.2 Results

There are no problem formulations for the k-means test case that are solved to optimality. All instances were stopped manually after 1800 seconds. The arc-flow formulation with sub-tour elimination constraints generates the best solution of \$14727, while the best bound is found by the standard arc-flow formulation, at \$12234. Combining these values a integrality gap of 16.9% is obtained. The lowest integrality gap found by single test instance was 18.2%. This integrality gap was generated by the arc-flow formulation with sub-tour elimination constraints.

Table 9: Results for K-Means test case

Instance	Running Time	Solution	Bound	Integrality Gap
24K-AL-S	1800s	23596	7103	69.9%
24K-AL-TW	1800s	22846	8735	61.2%
24K-AL-ST	1800s	19497	6833	65.0%
24K-AL-STA	1800s	19497	6762	65.3%
24K-AL-CI	1800s	23579	6175	73.8%
24K-AF-S	1800s	15529	12234	21.2%
24K-AF-TW	1800s	15510	12229	20.9%
24K-AF-ST	1800s	14727	12044	18.2%
24K-AF-STA	1800s	15578	12210	21.6%
24K-AF-CI	1800s	15346	12182	20.6%

Similar to the former test cases, also for the K-Means test case the arc-flow formulation provides superior solutions and bounds for every formulation improvement, when compared to the arc-load formulations.

Adding time windows improves the solutions found for both the arc-load and the arc-flow formulations. For the arc-load formulation the integrality gap decreases to 61.2%, compared to 69.9% for the standard formulation. For the arc-flow formulation the integrality gap decreases to 20.9%, compared to 21.2% for the standard formulation.

Both the split and aggregated sub-tour elimination constraints yields improved solutions and lower integrality gaps for the arc-load formulation. Both finds a solution of \$19 497, which is 17.4% less than the cost for the the standard formulation. For the arc-flow formulation only the formulation with split sub-tour elimination constraints is able to find an improved solution relative to the standard arc-flow formulation.

The formulations with clique inequalities gives a improved solution for the arc-load formulation, despite having a slightly inferior bound. Unlike from the production zones and the natural selection test cases, the k-means test case also produced a improved solution when clique inequalities where added to the arc-flow formulation.

The solution for routing for the arc-flow formulation with sub-tour elimination constraints is presented in figure 16. All clusters are served by self-operated vessels, so no external deliveries are needed. The vessel plotted with yellow arcs are one of the large vessel. This vessel serves the two northernmost clusters, but does not go back to the feed factory to reload feed. The other large vessel, plotted with gray arcs, reloads feed at the feed factory twice. This vessel serves nodes both south and north of the factory. The small vessels, plotted with green and purple, revisits the feed factory once each. The both serves nodes both south and north of the feed factory.

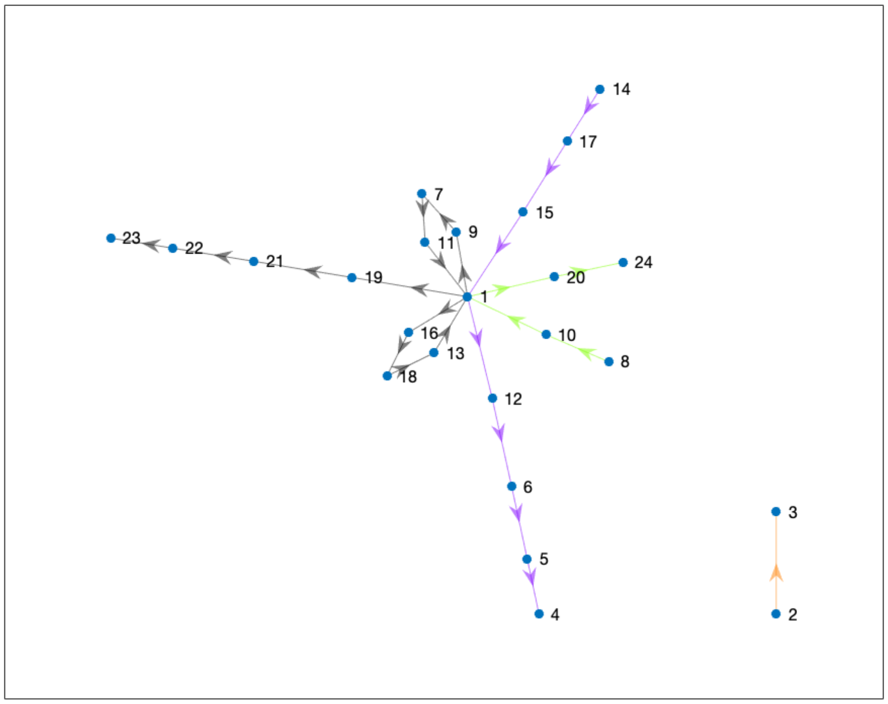

Figure 16: Routing for k-means test case. Nodes are denoted by node number. Node locations are arbitrary and does not indicate the real locations of the nodes. Ships are separated by arc colour.

9.7 Discussion of Test Case Results

The only test case able to solve the problem to optimality is the one with production zones. As expected, the integrality gaps increases with with an increasing number of nodes. The lowest integrality gap found for the test case with natural selection clustering was 8.7%, compared to 18.2% for the k-means clustering test case. Despite not being solved to optimality, the natural selection test case generated the best solution. The natural selection test case solved with the aggregated sub-tour elimination constraint found a solution of $11 519. This represents a 13.7% reduction compared to solutions found for the production zones test case. The higher number of clusters adds flexibility to the problem. With smaller cluster sizes, the smaller vessels can serve more clusters with only one delivery during the planning period. In addition, splitting a cluster into multiple smaller clusters allows vessels to serve only the cluster were feed requirements are the most urgent or suitable to the vessels current cargo levels, while other nodes can be served later by other vessels.

For all test cases the computational performance of the arc-load formulation is significantly improved when sub-tour elimination constraints or clique inequalities are added. For the production zones test case the computational time was decreased, while the larger test cases generated better solutions with the constraints added. The results for the arc-flow was less consistent, as the solution time was increased when adding sub-tour elimination constraints or clique inequalities for the production zones problem. However, the sub-tour elimination constraints also improved the solutions for the larger test cases when applied with the arc-flow formulation, though the effect was more limited than with the arc-load formulation.

Overall, the arc-flow formulations proves superior to the arc-load formulations in every aspect tested. There is no formulation improvement tested that provides better solutions and bounds with the arc-load, than with the arc-flow formulation. The arc-flow model has more continuous variables than the arc-load model, but the same number of binary variables and a lower number of constraints. The higher number of continuous variables does not seem to impact the running time of the model, as they do not add to the complexity of the search tree. However, the arc-flow formulation benefits from the lower number of variables and the lack of a formulation based on the Big M method. Ideally when solving a mixed-integer program, the linear relaxation should be as close to the convex hull of the union of all linear programs corresponding to fixing the integer variables to every possible value. As illustrated in figure 17, the linear relaxation underestimates the true cost when large Big M's are used, leading to loose bonds. The loose bounds make it harder to prune nodes based on objective value, thus more nodes have to be explored in the search tree, slowing down the process.

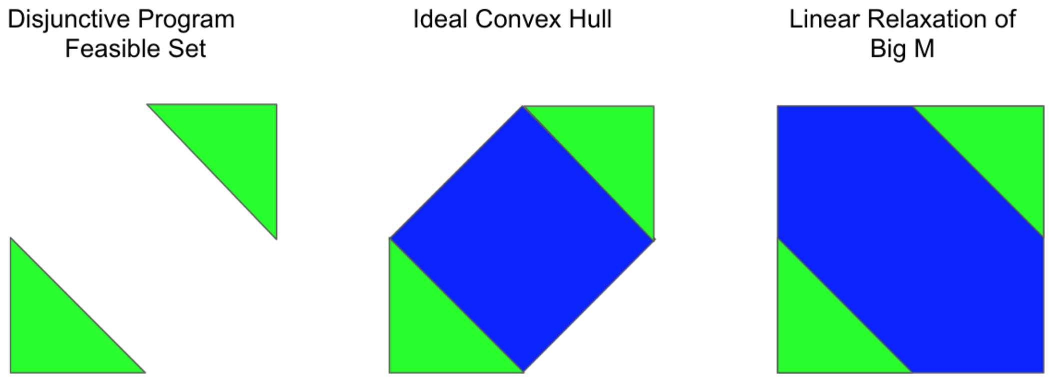

Figure 17: Illustration for bounds of a linear relaxation with big M constraints.

As the standard arc-flow formulation is simple and consistently generated good solutions and bounds, it is used in the following sections in order to test compatibility and consumption scenarios, as well as robustness.

9.8 Limiting compatibility between ships and harbours

As the results from the previous section clearly illustrates, the number of possible solutions and the computational complexity increases exponentially as the number of nodes increases. With larger problem sets, the optimal solution is never found, and the largest problem sets even fail to generate decent integrality gaps. In order to be able to increase the number of nodes, or reduce computational time, the number of possible solutions for routing of the ships can be decreased. This can be done by limiting which ship can sail to which port. In the given fleet, two ships are of smaller size and can carry 1500 tonnes of feed, while the other two ships are larger and can carry 3000 tonnes. The larger ships are also faster and can sail at a speed of 13 knots, compared to 10 knots for the smaller ones. The two larger ships are also more cost effective and environmentally friendly, as they run on LNG as compared to diesel for the two smaller ship. Thus the larger ships are better suited for serving fish farming locations far away from the Valnesset, Bjugn factory.

In order to test the scenarios with a limited compatibility a compatibility between ships and harbours, two compatibility matrices is generated for each test instance. The compatibility matrices determines whether a ship can visit a given harbour or not. The first matrix is referred to as a partly limited compatibility matrix. It is designed such that the smaller ships can only serve the harbours less than 200 nautical miles south of the factory and harbours less than 100 miles north of the factory. The larger vessel can only serve the the rest of the harbours, that is harbours more than 100 nautical miles north of the facotry and more than 200 miles south of the factory. To increase flexibility, ships can also serve one additional node south and north of their specified region.

The second matrix is referred to as a limited compatibility matrix. The matrix is designed like the partly limited compatibility matrix, but with additional restrictions. One of the large and one of the small vessels are restricted to only serve nodes north of the factory, while the remaining two vessels can only serve nodes south of the factory. Thus this matrix imposes stricter restrictions than the partly limited compatibility matrix.

Production Zones

Table 10: Results for compatibility scenarios with production zones test case

Instance	Compatibility	Running Time	Solution	Bound	Integrality Gap
12Z-AF-S	Unlimited	30s	13340	13340	0%
12Z-AF-S	Partly Limited	9s	13340	13340	0%
12Z-AF-S	Limited	4s	14734	14734	0%

The computational time is reduced with stricter compatibility restrictions. The limited compatibility problem is solved 7 times faster than the unlimited compatibility problem.

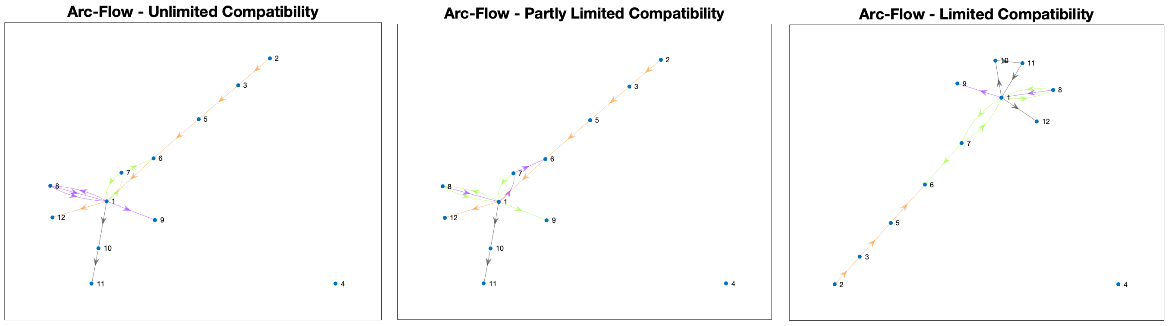
Arc-Flow - Unlimited Compatibility
Arc-Flow - Partly Limited Compatibility
Arc-Flow - Limited Compatibility

Figure 18: Routing for compatibility scenarios with production zones test case

The unlimited and partly limited compatibility solutions are similar, with both having vessel 1, plotted with yellow coloured arcs, serving the northernmost fish farms first before reloading feed at the factory and then serving the southernmost farm. Vessel 2, plotted with gray coloured arcs, serves only cluster 10 and 11, located south. The only differences between the solutions are that parts of the route for vessel 3 is changed with vessel 4. However, since vessel 3 and vessel 4 are modelled as identical and the same arcs are covered, the cost remains the same.

The solution of the limited compatibility problem differs more from the other two, as more arcs are prohibited. Vessel 1 is not able to serve cluster 12, like previously. Thus, cluster 12 have to be served by vessel 2. Vessel 2 have to reload feed before this cluster can served. As the arc back to the feed factory is longer than for vessel 1 than vessel 2, the cost is increased. The total cost for the limited compatibility problem is 10% higher than for the unlimited compatibility problem.

Natural Selection

Table 11: Results for compatibility scenarios with natural selection test case

Instance	Compatibility	Running Time	Solution	Bound	Integrality Gap
19N-AF-S	Unlimited	1800s	11533	10581	8.2%
19N-AF-S	Partly Limited	28s	20641	20641	0%
19N-AF-S	Limited	6s	21422	21422	0%

The computation times for the natural selection problem instance are reduced significantly by limiting compatibility. The problem is solved to optimality in 28 seconds with partly limited compatibility and 6 seconds with limited compatibility.

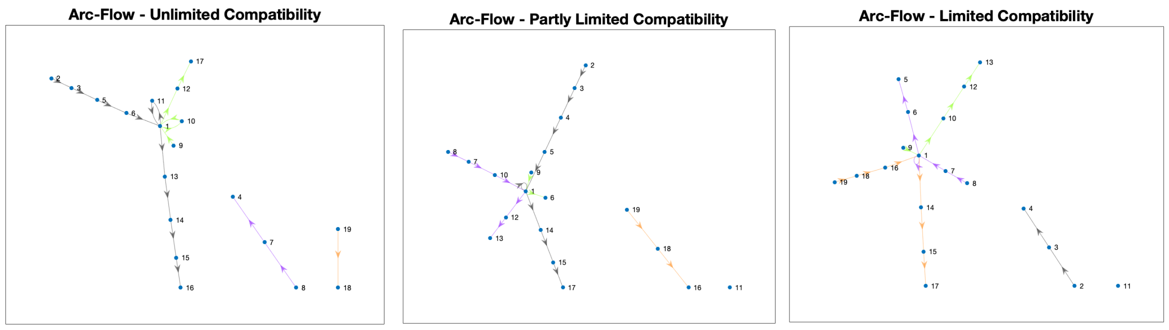
Arc-Flow - Unlimited Compatibility
Arc-Flow - Partly Limited Compatibility
Arc-Flow - Limited Compatibility

Figure 19: Routing for compatibility scenarios with natural selection test case

The cost for the problem with partly limited restrictions is 79% higher than for the unlimited compatibility problem. The cost is driven by the fact that cluster 11 have to be served by external vessels. In the unlimited compatibility problem, one of the larger vessels serves cluster 11. This is no longer an allowed solution in the partly limited compatibility problem, as this cluster have to be served by either one of the small vessels or external vessels. The total consumption at the cluster is equal to 2300 tonnes, which is more than the 1500 tonnes cargo capacity of the small vessels. Two trips are therefore needed to serve this node. Serving cluster 11 twice, while also serving the other clusters is not a feasible solution, and external deliveries are therefore required.

The same problem occurs in the limited compatibility problem, which have a cost that is 86% higher than the cost of the unlimited problem. Cluster 11 have to be served externally, as serving the cluster twice is not a feasible solution. This happens while one of the larger vessels is idle for 70% of the planning period. While the larger vessel have sufficient amount of time available to reload feed and serve cluster 11, this is not an allowed solution due to the compatibility restrictions. By modifying the limited compatibility matrix to allow the larger vessel to serve cluster 11, the cost is reduced to $12 906. This is only a 12% cost increase over the solution found with the unlimited compatibility matrix.

K-Means

Table 12: Results for compatibility scenarios with the K-Means test case

Instance	Compatibility	Running Time	Solution	Bound	Integrality Gap
24K-AF-S	Unlimited	1800s	15529	12234	21.2%
24K-AF-S	Partly Limited	120s	22287	22287	0%
24K-AF-S	Limited	32s	24319	24319	0%

The unlimited compatibility problem is able to solve the problem to an integrality gap of 21.2% in 1800 seconds. This compares to the partly limited and limited compatibility problems being able to solve the problem to optimality in 120 seconds and 32 seconds, respectively.

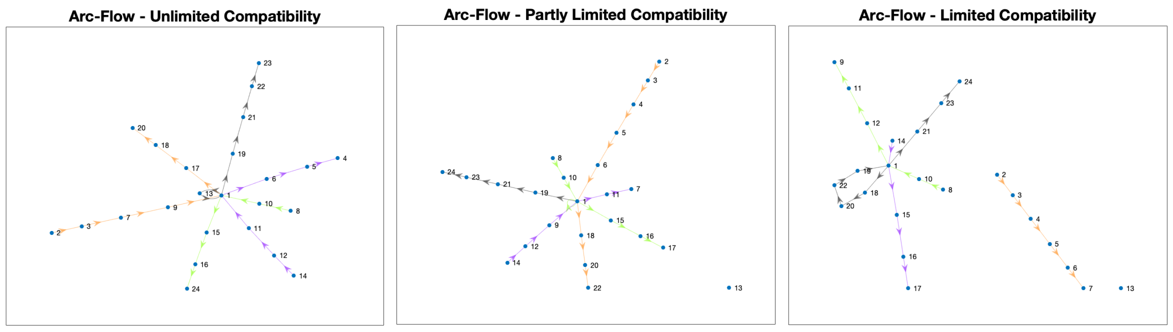
Arc-Flow - Unlimited Compatibility
Arc-Flow - Partly Limited Compatibility
Arc-Flow - Limited Compatibility

Figure 20: Routing for compatibility scenarios with k-means test case

The solution found for the unlimited compatibility problem is able so serve all clusters with feed from self-operated vessels. The largest vessels mainly serve the clusters furthest away from the factory, however one of the small vessels serve the southernmost cluster. The solution for the partly limited compatibility problem yields a cost that is 44% higher than the cost for the unlimited problem. The increase is mainly related to external deliveries. Cluster 13 have a demand of 1800 tonnes, which is more than the cargo capacity of a small vessel. As serving the node twice with one of the smaller vessels is not feasible, the cluster have to be served by externally.

The solution for the limited compatibility problem yields a cost that is 57% higher than the cost for the unlimited compatibility problem. Similar to the partly limited compatibility problem, cluster 13 is served by external vessels, significantly increasing the cost. Also less favorable arcs are used when sailing the vessels, due to stricter compatibility restriction. The most notable difference is that vessel 2 have to reload feed, instead of vessel 1. This increases the traveling length, and thus also cost. When modifying the limited compatibility matrix to allow the larger vessel to serve cluster 11, the cost is reduced to $17 775. This is only a 14% cost increase over the solution found with the unlimited compatibility matrix.

Discussion of Compatibility Scenarios

Computational times for the limited compatibility problems improves substantially when adding the limited ship to harbour compatibility restriction. The computational complexity is significantly reduced, as the number of decision variables are decreased. The limited compatibility reduces the number of variables that are indexed by both i and v. The greatest reduction of variables are seen among the x_{imjnv} and l_{imjnv} variables. There number of variables reduced are equal the these two variable sets. However, removing the binary x_{imjnv} variables reduces the number of possible branching variables. Thus, the reduction in computational complexity is more significant with the reduction of x_{imjnv} variables.

While reductions in ship to harbour compatibility can result in more than a 100 times improvement in computation time for problems of this size, they are also prone to increased costs. In the problem instances tested, the largest cost increases are due to external deliveries. The external deliveries are caused by some of the clusters having to be served multiple times by smaller vessels, as larger vessels are not allowed to serve these specific nodes. Multiple deliveries may not be a feasible solution, thus forcing the solution to include external deliveries. While it requires some knowledge about the specific optimization problem, the issue can easily be worked around by modifying the compatibility matrix to allow larger vessels to serve these nodes. Another alternative is to increase the number of possible arrivals to these clusters. By increasing the number of possible arrivals, vessels can discharge smaller quantities multiple times in order to form a feasible solution without external deliveries.

The test instance based on production zones clustering gives the smallest relative increase in costs with the limited compatibility restriction added. The production zones clustering gives a 10% increase in cost, compared to 86% and 57% for the natural selection clustering and the K-Means clustering respectively. The high increase in cost for the latter two was both driven by external vessels having to be utilized in order to meet consumption demands. By allowing larger vessels to serve one additional cluster of high demand, the relative cost increases were reduced to 12% and 14% respectively. This shows how small changes in the model formulations and restrictions, can impact the solutions objective value in large ways.

9.9 Consumption Scenarios

As the discussed in section 3.3, the consumption of feed varies considerably throughout the year. For Norway as a whole, the total consumption of fish feed usually peaks in July to September at around 50 000 tonnes a week. In February to March the total consumption is around 70% lower at 15 000 tonnes per week at it's lowest. Mowi only produces a fraction of the total fish feed supply in Norway. In Q1 2019 the total production at Valneset, Bjugn reached 61'000 tonnes. This equivalent to 4 700 tonnes a week. The total production for the year is estimated at around 300 000 to 350 000 tonnes. This is an average of about 7 000 tonnes a week, with peak consumption likely being around two times this number.

Three different consumption scenarios are generated and tested. In reality the consumption will vary between the different fish farms and some fish farms will not be in operation at all. These changes can be implemented in the input data files used for the model, either by setting consumption to zero for the given farms or by removing all associated variables and constraints for the non-operational farms. This would reduce the computational complexity of the problem. In the test cases these variations are neglected, and a equal consumption is assumed for all fish farms. The scenarios for the test cases can be found in the table below.

Table 13: Consumption scenarios

Scenario	Weekly consumption per farm	Weekly total consumption
Low	50 tonnes	7550 tonnes
Medium	100 tonnes	15100 tonnes
High	150 tonnes	22650 tonnes

It can be seen from the table that the medium consumption scenario of 15 100 tonnes a week is in line with the actual peak production at Mowi. Thus, the high consumption scenario of 22 650 tonnes a week must be considered a stress test, as the actual feed production is unlikely to reach these levels during normal operations. Unforeseen events, for example production halts, may lead to larger quantities having to be shipped when production is restarted. It is therefore of interest to look at scenarios with a higher demand than the actual peak consumption. The low consumption scenario places consumption at 7 550 tonnes, about 50% of the realistic peak demand. This is in line with the average production at Mowi's feed factory throughout the year.

All three consumption scenarios are tested with the three clustering methods found in earlier sections.

Production zones

The clustering by production zones are tested with the three consumption scenarios.

Table 14: Results for consumption scenarios with the production zones test case

Instance	Demand	Time	Solution	Bound	Int. Gap	Ext. Feed	Ext. Cost
12Z-AF-S	Low	26s	6788	6788	0%	-	-
12Z-AF-S	Medium	29s	13340	13340	0%	-	-
12Z-AF-S	High	1041s	23603	23603	0%	4200 t	9600

Table 15: Demand scenario results for production zones test instances

All consumption scenarios are solved to optimality, but the high consumption scenario have a computational time that is more than 30 times that of the low and medium consumption scenarios. The high consumption scenario is reliant on external vessels to serve some of the nodes. As there are multiple feasible combinations for which nodes that can be served by external vessels, branching trees are made for multiple of the decision variables determining external deliveries for a fish farm. Thus, the computational complexity increases significantly.

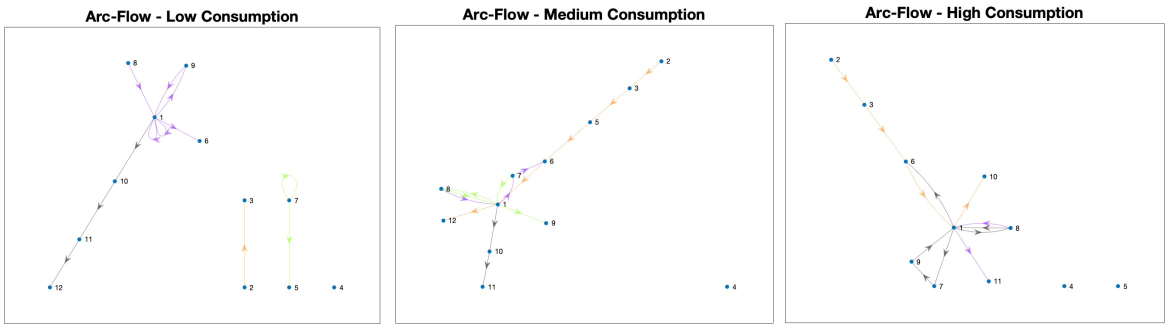
Arc-Flow - Low Consumption
Arc-Flow - Medium Consumption
Arc-Flow - High Consumption

Figure 21: Routing for consumption scenarios with production zones test case

The low consumption scenario gives a cost that is 49 % lower than the medium consumption scenario. The cost reductions are driven by the fact that only one of the vessels have to travel back to the factory, modelled as node 1, to get more feed. The other vessels are able to deliver feed to a high number of fish farms without loading more feed. None of the nodes have to be server multiple times, further reducing the sailing costs. The medium consumption scenario sees three of the ships going back to the factory to load more feed, while three of the fish farms are served twice. The vessels deliver a total of 15 910 tonnes of feed. The high consumption scenario is 77% more costly than the medium consumption scenario. The increase in costs are mainly due to $9 600 associated with external deliveries. A total of 18 610 tonnes are delivered with self-operated vessels, while 4 200 tonnes are delivered by external vessels to two clusters.

Natural Selection

The clustering by natural selection are tested with the three consumption scenarios.

Table 16: Results for consumption scenarios with the natural selection test case

Instance	Demand	Time	Solution	Bound	Int. Gap	Ext. Feed	Ext. Cost
19N-AF-S	Low	11s	5038	5038	0%	-	-
19N-AF-S	Medium	1800s	11533	10581	8.2%	-	-
19N-AF-S	High	1800s	32579	28005	14.0%	9150 t	21350

The low consumption scenario is solved to optimality in 11 seconds, while the medium and high consumption scenarios are solved with integrality gaps of 8.2% and 14.0% respectively. This follows the same trend as seen with the test instances based on production zones. The high consumption scenario requires external vessels in order to handle demand at fish farms, and multiple combinations of which fish farms can be served externally are feasible and close to optimality. Thus, complex branching trees are made for each of the decision variables for external deliveries.

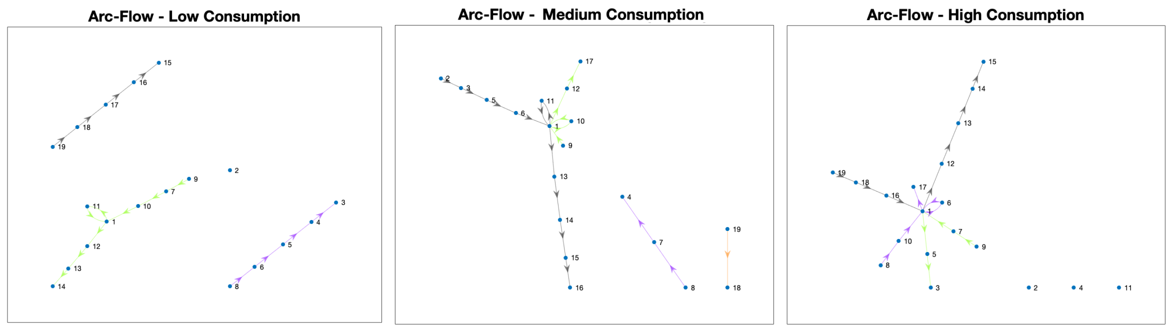
Arc-Flow - Low Consumption
Arc-Flow - Medium Consumption
Arc-Flow - High Consumption

Figure 22: Routing for consumption scenarios with natural selection test case

The low consumption scenario yield a cost that is 56% lower than the medium consumption cost. The cost reduction is driven by the vessels ability to serve multiple fish farms without reloading feed. Only one of the small vessels have to revisit the feed factory. One of the large vessels is only serving node 2, limiting travel costs. The medium consumption results have two of the vessels needing to reload feed twice, while two other vessels do not load more feed at the factory during the planning period. The high consumption scenario cost found is 182% above that of the medium consumption scenario. This is despite sailing costs of self-operated vessels being lower. Thus, the difference is driven by a \$21 350 cost associated with external deliveries. A total of 5 harbours are served with external deliveries.

Table 17: Results for consumption scenarios with the K-Means test case

Instance	Demand	Time	Solution	Bound	Int. Gap	Ext. Feed	Ext. Cost
24K-AF-S	Low	422s	7454	7454	0%	-	-
24K-AF-S	Medium	1800s	15529	12234	21.2%	-	-
24K-AF-S	High	1800s	24447	21453	12.2%	4500 t	10500

K-Means

The clustering by the k-means method is tested with three different consumption scenarios. The low consumption scenario is solved to optimality in 422 seconds. The medium and high consumption scenarios are not solved to optimality, but is stopped manually after 1800 seconds. Their respective integrality gaps are 21.2% and 12.2%. Unlike the test case natural selection, the k-means solves the high consumption to a smaller integrality gap than the medium consumption scenario.

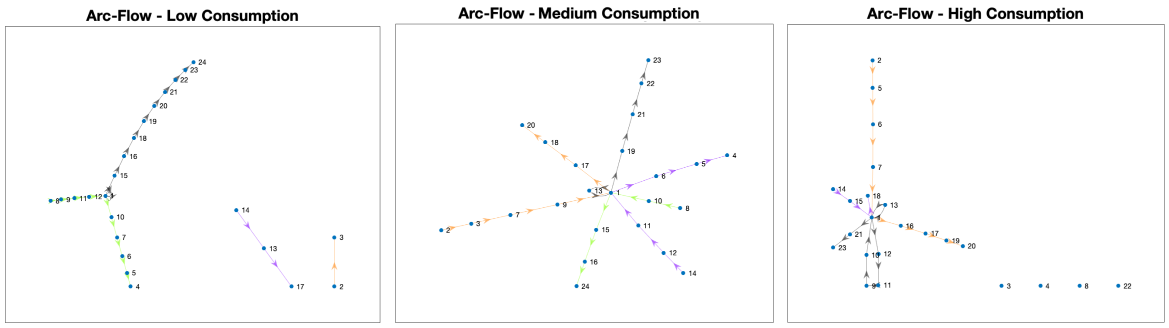
Arc-Flow - Low Consumption
Arc-Flow - Medium Consumption
Arc-Flow - High Consumption

Figure 23: Routing for consumption scenarios with k-means test case

The solutions generated for the low and medium consumption scenarios only have self-operated vessels serving nodes. The low consumption scenario allows for less costly arcs to be sailed, reducing the cost by 52% relative to the medium consumption scenario. While the low consumption scenario only have two vessels visiting the feed factory harbour, all four vessels reload feed at the feed factory for the medium consumption scenario. The solution for the high consumption scenario have a cost that is 57% more than the cost of the medium consumption test case. Like the production zones and natural selection test cases the main cost driver is the external deliveries.

Discussion of Consumption Scenario Results

While the test instance based on production zones clustering with 12 harbours is the only one to solve all consumption scenarios to optimality, the natural selection clustering with 19 nodes finds the best solution for both the low and medium consumption scenarios. This is despite not being able to solve the latter to optimality.

For the low consumption scenario the natural selection clustering yields a cost that is 26% lower than the cost given with the production zones clustering and 32% lower than with the K-Means clustering method. The natural selection clustering is the only method able to find a solution were only one of the vessels have to reload feed. Neither of the clustering methods yields solutions were external deliveries are utilized for the low consumption scenario.

Also for the medium consumption scenario, the natural selection clustering yields the lowest cost. The cost is 14% lower than the optimal cost found with the production zones clustering and 26% lower the cost found using the K-Means clustering method. Neither of the clustering methods yields solutions were external deliveries are utilized for the medium consumption scenario.

For the high consumption scenario, the production zones clustering yields the lowest cost. The cost is 27% lower than the optimal cost found with the natural selection clustering and 3% lower the cost found using the K-Means clustering method. All of the clustering methods yields solutions were external deliveries are utilized for the medium consumption scenario.

9.10 Robustness

When talking about a robustness of a system, it refers to the systems ability to tolerating perturbations. Perturbations can happen in multiple ways for a feed delivery system. This includes changes in consumption due to illness or temperature changes, delayed deliveries to to weather or vessel failures, or not being able to make deliveries at all due to unexpected feed factory maintenance. In order to be able to tolerate such perturbations, the system have to be designed with some form of slack or extra capabilities that can be utilized when needed.

In the fish feed distribution problem, the slack can be implemented either in the form of extra time between deliveries or by adding restrictions for minimum inventory levels. The most serious situation and the largest implicit costs are associated with fish farms running out of feed. If fish farms are running out of feed, the fish stops growing, which reduces the revenues for the aquaculture company. If a delivery is delayed, extra feed inventory can be used to feed the fish while waiting for the delivery. In cases of cancelled deliveries, extra feed inventory gives operators time to replan other delivery methods. Thus, adding restrictions for minimum inventory levels is an appropriate way of increasing the robustness of the system.

Minimum Inventory Restrictions

All problem formulations presented earlier are modelled with a minimum inventory level of 0 and a minimum inventory level at the end of the planning period equal to 40% of the inventory capacity. Thus, the minimum inventory can go to zero multiple times during the planning period, as long as the feed inventory is refilled immediately and the inventory levels at the end of the planning period is equal to at least 40%.

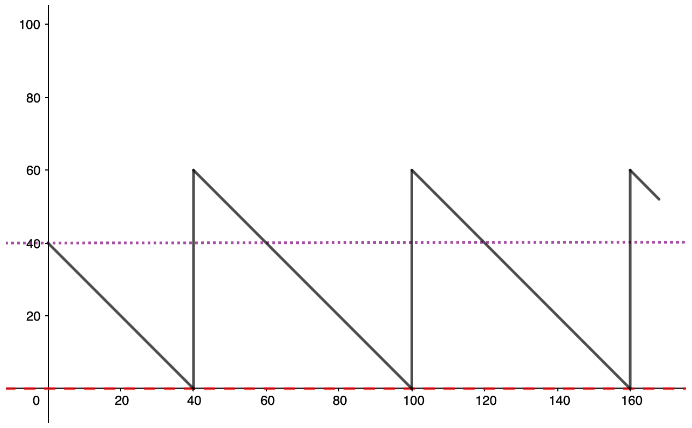

Figure 24: Illustration for minimum inventory level of 0% and minimum inventory at the end of the planning period of 40%.

Higher levels for minimum inventory can be applied in order to increase the robustness of the model. While the increased robustness limits the effects of unforeseen events, the higher levels for minimum inventory also forces the first arrival to happen earlier in the planning period. This can increase costs, or may not even be feasible.

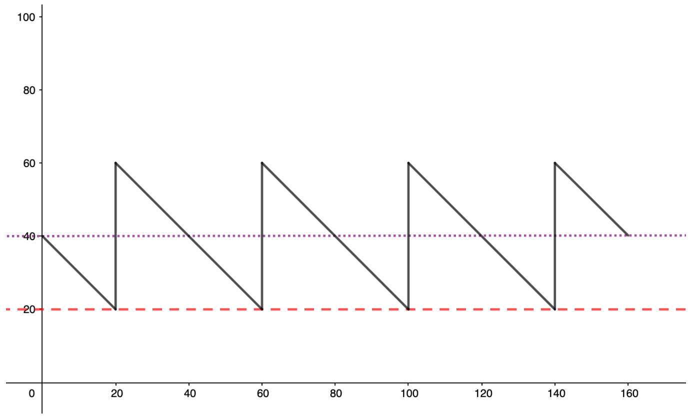

Figure 25: Illustration for minimum inventory level of 20% and minimum inventory at the end of the planning period of 40%.

Results

Minimum inventory levels between 0% and 20% of cluster capacity is tested. The 12 harbour production zones clustering method is used as the underlying test instance.

Table 18: Results for minimum inventory levels with the production zones test case

Instance	Minimum Inventory	Running Time	Solution	Bound	Integrality Gap
12Z-AF-S	0%	30s	13340	13340	0%
12Z-AF-S	5%	32s	13340	13340	0%
12Z-AF-S	10%	30s	13340	13340	0%
12Z-AF-S	15%	280s	14019	14019	0%
12Z-AF-S	20%	842s	15356	15356	0%

The first three test instances of 0%, 5% and 10% minimum inventory all yields the same solution. The test instance of 15% minimum inventory have a costs that is 5.1% higher than the costs of the 0% minimum inventory test instance. The cost increase is attributable to changes in the arcs that are sailed. The main difference is that one large and one small vessel serves cluster 7, as compared to two small vessels serving the cluster for the 0% minimum inventory test instance.

The test instance of 20% minimum inventory have a costs that is 15.1% higher than the costs of the 0% minimum inventory test instance. Also here, the cost is attributable to changes in the arcs that are sailed. One of the small vessels serves cluster 11, as compared to one of the large vessels serving the cluster for the 0% minimum inventory test instance. Cluster 8 is served by both a large and a small vessel, as compared to being served twice by small vessels for the 0% minimum inventory test instance.

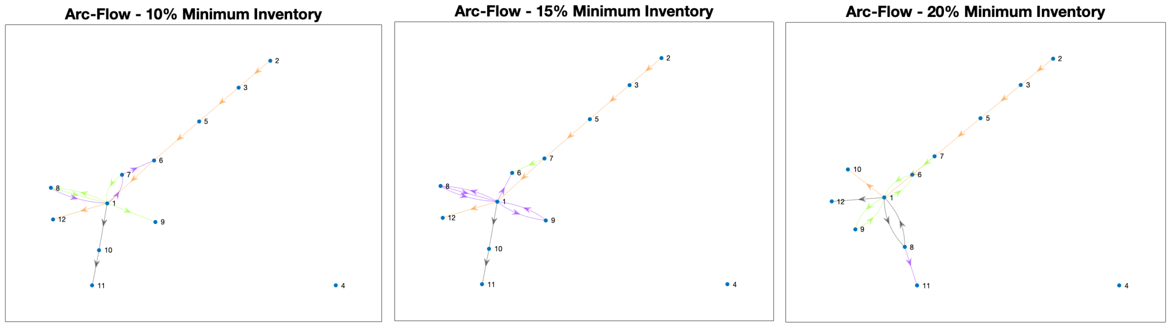
Arc-Flow - 10% Minimum Inventory
Arc-Flow - 15% Minimum Inventory
Arc-Flow - 20% Minimum Inventory

Figure 26: Routing for minimum inventory scenarios with production zones test case

Discussion of Minimum Inventory Restrictions

The results show how a modest increase in costs can ensure much higher minimum feed inventory levels. As all test cases in this section have included a minimum inventory level at the end of the planning period of 40%, the total amount of feed that have to be delivered is equal for all test cases. However, the added minimum inventory level restrictions during the planning period force deliveries to take place earlier. Thus, more costly routes have to be used in order to serve all clusters in time.

While it was not the case for any of the test instances in this section, higher minimum inventory levels can also lead to a higher number of arrivals being necessary at each cluster. This happens when the difference between minimum and maximum inventory levels are small, limiting the total amount of feed that can be discharged at each cluster for each arrival. Another reason for a increased number of arrivals is that only a small amount of feed is discharged at each cluster at the first arrivals, in order to save to time, so that other clusters can be served before they run out. The model formulations in thesis have assumed a discharge speed of 250 tonnes per hour. This makes the time used for discharging feed relatively small, as compared to the total sailing time.

10 Discussion

Many different formulations can be applied in order to optimize the use of the fleet of fish feed carriers. While no model can include all factors in a complex real life system, a good model is able to simulate the most important aspects of a real life operation and generate useful outputs that can improve operations.

The thesis by Haugland and Thygesen (2017) was used as a starting point for this thesis. While Haugland and Thygesen formulated a VRP solved by a two-stage heuristic, an IRP formulation have been considered in this thesis. Both approaches have their advantages and disadvantages.

Haugland and Thygesen's VRP formulation are able to solve problem instances with up to 24 nodes to optimality within seconds. The 24 harbour arc-flow IRP formulation applied in this thesis is only able to get to an integrality gap of about 20% within 30 minutes, if no compatibility restrictions are imposed. Models occasionally have to be rerun in order to adapt to real life changes. These changes may include fish being moved to other fish farms, sudden fish death due to algae, bacteria or viruses or weather changes that increases sailing times. When sudden changes like these happens, the planning models will have to be able to deliver useful outputs rapidly, so that ships do not lie idle waiting for instructions on where to sail. In time critical situations, complex IRP formulations are less suitable than the VRP.

The computational study shows how quickly the complexity and solution time increases with additional nodes in the IRP. As the IRP problem can potentially include multiple nodes for each harbour in order to resemble separate arrivals, the number of possible arcs grows much faster for the IRP formulation, than the VRP formulation. Thus, model simplifications are more essential in the modelling of a IRP. The model simplifications introduced helps improve the computational time, but it also limits the model, as real world implications are neglected.

A continuous time model does not allow for variable consumption rates. As the planning horizons used in the problem is not longer than two weeks, deviations in sea temperature and thus consumption will likely be small. Day-to-day variations in feed consumption are likely to cancel out each other. Also the problem can be modelled with stricter inventory level restrictions, adding a margin of safety before a fish farming location runs out of inventory. Thus, the modelling the problem with a constant consumption rate can be seen as reasonable simplification.

Deterministic modelling does not include stochastic variables, and therefore also neglects changing weather, unforeseen changes in feed consumption, failures on ships and other factors that may impact operations. While modelling stochastic behaviour is hard and increases the computational complexity of the model in it self, gathering and applying the correct data in terms of probability distributions for different failures is a even larger challenge. Without trustworthy probability distributions, stochastic modelling may prove to be of little value in planning of operations. However, knowing that the modelling does not incorporate stochastic behaviour a conservative approach should be used when setting the feed level restrictions.

Single product and single compartment modelling are other simplifications made. In real operations different fish farms would need different feed based on the size, health and growth stage of the fish. Realistically different products would be shipped in different compartments on the ships. Adding multiple products and compartments to the model would increase the computational complexity significantly and is therefore neglected.

Other simplifications that are tested includes ship and harbour compatibility. By limiting which ship can travel where, the number of arcs are decreased significantly and the model complexity is sharply decreased. The downside to limiting the ship and harbour compatibility is that more efficient routes are neglected. However, often the more efficient routes are never found when limitations to compatibility is not included, as the formulations without limitations can not be solved to optimality. The limitations to compatibility may also be applied to model actual restrictions that limits ships to visit given fish farms.

In addition to the simplifications discussed, model formulation changes and formulation improvements have also been considered in order to improve computational time. Among the different model formulations, the arc-flow model consistently proved superior to other model formulations, as it gave better bounds and solutions than alternative formulations. For the smaller test instances that were solved to optimality, the arc-flow showed the shortest running time. While the arc-flow formulation incorporates more variables than the arc-load formulation, it has the benefit of less constraints. Quadratic constraints that are linearized in the arc-load formulation, is also avoided in the arc-flow formulation.

Formulation improvements includes bounds that can provide better solutions faster, but does not impact the value of the optimal solution of a problem. Model improvements that are discussed includes split and aggregated sub-tour elimination constraints and clique inequalities. Both the split and aggregated sub-tour elimination constraints saw large improvements when implemented with the arc-load formulation. However, their effect when implemented with the arc-flow model was limited.

In addition to the other formulation improvements, tightening of time windows was implemented. Unlike the other formulation improvements, tightening of time windows may impact the optimal solution. Only simple tightening of time windows were considered, with arrivals limited to the first six days of the planning period. The tightening of time windows did not show any improvements in integrality gaps.

By limiting ship to harbour compatibility the computational complexity was reduced significantly. While the testing of the compatibility restrictions was only an example of how these restrictions could be used, fish farming companies can make modification for the compatibility matrices suiting their own need. While the compatibility matrices generated for the test cases were based on splitting the vessels from north to south, fish farming companies may also have other restrictions. These restrictions can include harbours not being compatible with the feed discharge system onboard a vessel or a large vessel not being able to serve farms in shallow fjords. In the test cases it was seen how some clusters with large demands can not feasibly be served by smaller vessels. When implementing compatibility matrices, such factors have to be taken into consideration in order to avoid large cost increases. In the formulated problem the cost increases are largest when clusters have to be served by external farms. This is avoided by letting larger vessels serve the farms of highest consumption, or by allowing a high number of arrivals by smaller vessels.

The consumption scenarios tested indicated that delivery costs increase with higher demands. This is expected, as the vessels have to reload feed at the factory more often. All high consumption scenarios tested indicated that external deliveries have to be used in order meet feed requirements at all clusters. While this leads to very high costs, it should be noted that the high consumption scenario that is tested is an extreme case, and is about 50% above the actual total peak consumption for Mowis fish farms. Neither of the low or medium consumption scenarios tested had to utilize external deliveries.

The model is formulated in order to avoid end of horizon effects, by adding restrictions for feed levels at the end of the planning period. In all test cases the minimum feed inventory at harbours at the end of the planning period have been required to be equal to or greater than that of the start of the planning period. While fish farms are allowed to have lower inventory levels during the planning period, the restrictions for the end of planning period inventory levels ensures that the next planning period is not starting with empty inventory levels. These end of horizon restrictions may lead to elevated costs. An alternative approach that can be considered is lowering the feed inventory level restriction at the end of the planning period, and instead add a constraint for the average inventory levels at the end of the planning period. This could lead to lower costs, while at the same time limiting feed shortage lags from prior planning periods.

While feed inventory levels are implemented in the model, some end of horizon effects that are not accounted for. This includes the ship cargo levels and ship locations. The optimal solutions found often have ships with low cargo levels at the end of the planning horizon. To some extent this is compensated by harbours having higher feed inventory levels than at the start of the planning period. However, a better solution would include restrictions to limit these end of horizon effects as well. The current model formulation allows to fix given harbours as ending nodes, however additional restrictions would have to be added in order to include ship cargo levels at the end of the planning period.

It theory the current model formulations could be utilized for longer term planning. However, as the computational time increases exponentially with a increasing number of possible visits per node, routing and scheduling could only optimized for limited time horizon. While the end of horizon effects can be avoided by additional constraints, they may not necessarily be efficient in terms of cost. Instead a rolling horizon heuristic or other more sophisticated methods should be implemented in order to utilize the model for long term planning.

While parallel computing was discussed in the literature study, different approaches for parallel computing was not evaluated in the thesis. However, it should be noted that commercial solvers incorporate parallel computing when solving integer problems. The fact that parallel computing is incorporated means that significant improvements in computational time can likely be achieved by utilizing specialized computers. As the model only needs to be solve at the order of once a week, cloud computing would likely prove effective when running the model. Deploying the model to a cloud based system would add the possibility of only having to pay for the computational capacity when the model is run. At the same time the computational capacity can be scaled up when needed in order to improve computational time.

11 Conclusion

Being able to deliver feed in a safe and efficient way is important in order to lower cost in the salmon farming industry. Mowi are achieving cost effective feeding by a vertically integrated supply chain, where they own and operate a feed factory that produces most of the feed consumed at their fish farm. In this thesis efficient distribution of feed have been the discussed and an Inventory Routing Problem (IRP) have been formulated.

The IRP formulation was selected over Vehicle Routing Problem (VRP) formulations, as the benefits of managing routing and inventory levels simultaneously are significant. The IRP was modelled with three different model types, including an arc-load model, an arc-flow model and a Dantzig-Wolfe decomposition. The models included transportation with ships in a given fleet, and the possibility of hiring external vessels. Varying consumption rates, implementation of stochastic parameters and multiple products and compartments were considered, but not included in the model, due to increased computational complexity.

Three test cases based on different clustering methods were tested with the Arc-Load and the Arc-Flow model formulations. The test cases comprised of 12, 19 and 24 clusters, where each cluster was modelled as a harbour with additional internal sailing times in the problem formulations. The smallest test case was the only one solved to optimality without adding compatibility restrictions. The Arc-Flow model consistently solved this test case to optimality faster than the other models, regardless which formulation improvements were added. The larger test cases were not solved to optimality, but integrality gaps of 8.2% and 20.2% were found for the 19 harbour and 24 harbour test cases respectively. The Arc-Flow model continued to perform better than the other models on the larger test cases, generating both better solutions and bounds.

Formulation improvements that were tested includes tightening time windows, sub-tour elimination constraints and clique inequalities. When applied with the Arc-Load model, the sub-tour elimination constraints and the clique inequalities reduced computational time for the small test instances and generated improved solutions and bounds for the larger test instances. The effect of the formulation improvement was more limited when used with the Arc-Flow model. The sub-tour elimination constraint gave a slightly better solutions for the 19 harbour test case, but performance was decreased with the smaller test case.

Limiting ship to harbour compatibility generated solution solved to optimality multiple times faster than without limited compatibility. The larger test cases gave significantly increased costs when adding compatibility restrictions, but it was shown how a major part of the cost increases could be avoided by small changes to the compatibility restrictions. These changes includes allowing the largest vessels to serve the clusters with highest feed demands.

Robustness of the model was improved by adding stricter minimum levels for feed inventory at fish farms. Minimum levels for feed inventory up to 10% of inventory capacity did not result in increased cost. The solutions generated with minimum levels at 15% and 20% of inventory capacity changed somewhat, but resulted in only minor increases in costs.

12 Further work

Further work includes both potential formulation improvements and implementation and testing of new solution methods. At any given time it is essential that the model is able to capture and optimize the key components of the system it is meant to optimize. While the current problem formulation includes both inventory management, routing and the possibility of external hiring in order to minimize cost, it is also possible to add more elements as penalty costs for serving locations too late. By allowing late deliveries and adding a penalty cost, better routing alternatives can be found that lowers the total cost.

As IRPs of this size represents great complexity, the most important improvement may be related to better solution methods. Implementation of heuristics and meta-heuristics in order to faster obtain feasible solutions should be considered for further work. Another way of solving the problem fast, can be achieved by implementing parallel computing algorithms. As already shown in section 4.6 , there are multiple ways of implementing parallel computing with the branch-and-bound algorithm. Parallel computing is already implemented in commercial solvers, as Gurobi, therefore a parallel computing algorithms will have to be highly efficient in order to provide results faster than the commercial solvers. Interesting approaches may include finding algorithms that utilizes the structure of the problem, in order to solve it faster.

Operating with live fish in harsh and rapidly changing climates makes the problem prone to great uncertainties. While the consumption of fish feed and travelling time between clusters are modelled as deterministic, a more robust model could implement stochastic variables in order to account for the uncertainties in the model. Implementation would however also increase computational complexity, further increasing computational time or restricting the feasible size of the test cases.

While the fish farms are modelled as clusters in this thesis, limited work has been done modelling the operations internally in the cluster. In order to obtain more accurate solutions, optimization of the travelling times internally in the clusters should be carried out. If the current estimates for the travelling times internally are underestimated, the routing may not be feasible in real life and cause delays in deliveries relative to the operation plan. In the worst case this could lead to a decline in the growth of the fish, reducing revenues for the salmon farming company.

If models like the ones presented in this thesis are to be implemented in an industrial setting, routines for measuring and gathering data will have to be developed. The value of the results generated by the model is highly dependent on the accuracy of the input data. Especially metrics like inventory levels, vessel cargo levels and vessel positions at the start of the planning period are crucial in order for the model to generate useful results. While harder to predict, accurate predictions for feed consumption is also one of the important input metrics. As the feed consumption changes throughout the year, implementing routines for continuously updating these metrics is an absolute necessity if the model is to be implemented.

References

Turid Synnøve Aas, Maike Oehme, Mette Sørensen, Gaojie He, Ingolf Lygren, and Torbjørn Åsgård. Analysis of pellet degradation of extruded high energy fish feeds with different physical qualities in a pneumatic feeding system. *Aquacultural engineering*, 44 (1):25–34, 2011.

Bjørn Ådlandsvik. Forslag til produksjonsområder i norsk lakse-og ørretoppdrett. 2015.

Yossiri Adulyasak, Jean-François Cordeau, and Raf Jans. Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4):851–867, 2015.

Agostinho Agra, Marielle Christiansen, and Alexandrino Delgado. Mixed integer formulations for a short sea fuel oil distribution problem. *Transportation Science*, 47(1): 108–124, 2013.

Agostinho Agra, Marielle Christiansen, Alexandrino Delgado, and Luidi Simonetti. Hybrid heuristics for a short sea inventory routing problem. *European Journal of Operational Research*, 236(3):924–935, 2014.

Agostinho Agra, Marielle Christiansen, Kristine S Ivarsøy, Ida Elise Solhaug, and Asgeir Tomasgard. Combined ship routing and inventory management in the salmon farming industry. *Annals of Operations Research*, 253(2):799–823, 2017.

Aktieselskabet Schouw & Co. Årsrapport 2017, biomar. page 22, 2018. URL `https://www.schouw.dk/investorer/regnskaber-og-praesentationer/`.

Claudia Archetti, Luca Bertazzi, Gilbert Laporte, and Maria Grazia Speranza. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41(3):382–391, 2007.

Erland Austreng, Trond Storebakken, and Torbjørn Åsgård. Growth rate estimates for cultured atlantic salmon and rainbow trout. *Aquaculture*, 60(2):157–160, 1987.

Walter J Bell, Louis M Dalberto, Marshall L Fisher, Arnold J Greenfield, Ramchandran Jaikumar, Pradeep Kedia, Robert G Mack, and Paul J Prutzman. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, 13(6):4–23, 1983.

Aslak Berge. *These are the world's 20 largest salmon producers.* Salmonbusiness AS, 2017. URL `https://salmonbusiness.com/these-are-the-worlds-20-largest-salmon-producers/`.

Luca Bertazzi and M Grazia Speranza. Matheuristics for inventory routing problems. In *Hybrid Algorithms for Service, Computing and Manufacturing Systems: Routing and Scheduling Solutions*, pages 1–14. IGI Global, 2012.

Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.

JR Brett, TDD Groves, et al. Physiological energetics. *Fish physiology*, 8(6):280–352, 1979.

Ann Campbell, Lloyd Clarke, Anton Kleywegt, and Martin Savelsbergh. The inventory routing problem. In *Fleet management and logistics*, pages 95–113. Springer, 1998.

Cermaq. Annual report 2015,cermaq's supply chain. 2016. URL `https://www.cermaq.com/wps/wcm/connect/cermaq/cermaq/our-company/annual-report/article-annual-reports/gri-index/cermaqs-supply-chain`.

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

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

Leandro C Coelho, Jean-François Cordeau, and Gilbert Laporte. Thirty years of inventory routing. *Transportation Science*, 48(1):1–19, 2013.

BGW Craenen, AE Eiben, and E Marchiori. How to handle constraints with evolutionary algorithms. *Practical Handbook Of Genetic Algorithms: Applications*, pages 341–361, 2001.

George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.

George B Dantzig and Philip Wolfe. Decomposition principle for linear programs. *Operations research*, 8(1):101–111, 1960.

Stéphane Dauzère-Pérès, Atle Nordli, Asmund Olstad, Kjetil Haugen, Ulrich Koester, Myrstad Per Olav, Geir Teistklub, and Alf Reistad. Omya hustadmarmor optimizes its supply chain for delivering calcium carbonate slurry to european paper manufacturers. *Interfaces*, 37(1):39–51, 2007.

Harald Ellingsen and Svein Aanond Aanondsen. Environmental impacts of wild caught cod and farmed salmon-a comparison with chicken (7 pp). *The International Journal of Life Cycle Assessment*, 11(1):60–65, 2006.

SE Elmaghraby and MK Wig. On the treatment of stock cutting problems as diophantine programs. *Operations Research Report*, 61, 1970.

JR Evans. Solving multicommodity transportation problems through aggregation. In *ORSA/TIMS Joint National Meeting, Los Angeles (November)*, 1978.

E. Moe EY. The norwegian aquaculture analysis 2017. *The Norwegian aquaculture analysis*, 2017.

K. Fagerholt. Optimal fleet design in a ship routing problem. 1999.

Kayvon Fatahalian. From shader code to a teraflop: How gpu shader cores work. In *ACM SIGGRAPH*, 2010.

Matteo Fischetti and Domenico Salvagnin. Feasibility pump 2.0. *Mathematical Programming Computation*, 1(2-3):201–222, 2009.

Matteo Fischetti, Fred Glover, and Andrea Lodi. The feasibility pump. *Mathematical Programming*, 104(1):91–104, 2005.

FishChoice Inc. *Fishchoice Sourcing Summary*. FishChoice Inc., 2018. URL `https://fishchoice.com/buying-guide/atlantic-salmon`.

Fisheries, FAO. Aquaculture department. 2013. *Global Aquaculture Production Statistics for the year*, 2011.

Sveinung Fivelstad, Rune Waagbø, Sigurd Stefansson, and Anne Berit Olsen. Impacts of elevated water carbon dioxide partial pressure at two temperatures on atlantic salmon (salmo salar l.) parr growth and haematology. *Aquaculture*, 269(1-4):241–249, 2007.

Bernard Gendron and Teodor Gabriel Crainic. Parallel branch-and-branch algorithms: Survey and synthesis. *Operations research*, 42(6):1042–1066, 1994.

Ralph E Gomory. An algorithm for integer solutions to linear programs. *Recent advances in mathematical programming*, 64:260–302, 1963.

Gurobi Optimization. Gurobi documentation - varhintval. 2018. URL `http://www.gurobi.com/documentation/8.1/refman/varhintval.html`.

M. G. Haugland and S. Thygesen. Use of clusters in a route generation heuristic for distribution of fish feed. 2017.

Yaowu Hu, Hong Shang, Haowen Tong, Olaf Nehlich, Wu Liu, Chaohong Zhao, Jincheng Yu, Changsui Wang, Erik Trinkaus, and Michael P Richards. Stable isotope dietary analysis of the tianyuan 1 early modern human. *Proceedings of the National Academy of Sciences*, 106(27):10971–10974, 2009.

JE Juell, DM Furevik, and Å Bjordal. Demand feeding in salmon farming by hydroacoustic food detection. *Aquacultural Engineering*, 12(3):155–167, 1993.

Jin Hyuk Jung and Dianne P O'Leary. Cholesky decomposition and linear programming on a gpu. *Scholarly Paper, University of Maryland*, 2006.

Kontali Analyse. The salmon farming industry in norway 2018. 2018.

KG Kumar. Samudra report no. 28, april 2001. 2001.

Ailsa H Land and Alison G Doig. An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, pages 497–520, 1960.

Victor W Lee, Changkyu Kim, Jatin Chhugani, Michael Deisher, Daehyun Kim, Anthony D Nguyen, Nadathur Satish, Mikhail Smelyanskiy, Srinivas Chennupaty, Per Hammarlund, et al. Debunking the 100x gpu vs. cpu myth: an evaluation of throughput computing on cpu and gpu. *ACM SIGARCH computer architecture news*, 38(3): 451–460, 2010.

Olav Lekve. Norsk oppdrettsnæring. Fiskeridirektoratet, 2016. URL `https://www.barentswatch.no/artikler/Norsk-oppdrettsnaring/`.

Huimin Lu, Yujie Li, Min Chen, Hyoungseop Kim, and Seiichi Serikawa. Brain intelligence: go beyond artificial intelligence. *Mobile Networks and Applications*, 23(2):368–375, 2018.

James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

Marine Harvest. Salmon farming industry handbook 2018. 2018, 2018a.

Marine Harvest. Våre norske regioner, marine harvest. 2018b. URL `http://marineharvest.no/about/vare-regioner/`.

Kurt Mehlhorn and Peter Sanders. *Algorithms and data structures: The basic toolbox*. Springer Science & Business Media, 2008.

Rune Melberg and Reggie Davidrajuh. Modeling atlantic salmon fish farming industry: freshwater sub model simulation. In *Proceedings of the 2009 Spring Simulation Multiconference*, page 38. Society for Computer Simulation International, 2009.

Nenad Mladenović and Pierre Hansen. Variable neighborhood search. *Computers & operations research*, 24(11):1097–1100, 1997.

Siti Nurminarsih, Ahmad Rusdiansyah, Nurhadi Siswanto, and Anang Zaini Gani. Dynamic-inventory ship routing problem (d-isrp) model considering port dwelling time information. *Procedia Manufacturing*, 4:344–351, 2015.

Siti Nursyahida Othman, Noorfa Haszlinna Mustaffa, Nor Haizan Mohamed Radzi, Roselina Sallehuddin, and Nor Erne Nazira Bazin. A modelling of genetic algorithm for inventory routing problem simulation optimisation. *International Journal of Supply Chain Management*, 5(4):43–51, 2016.

Manfred Padberg and Giovanni Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM review*, 33(1):60–100, 1991.

Dražen Popović, Milorad Vidović, and Gordana Radivojević. Variable neighborhood search heuristic for the inventory routing problem in fuel delivery. *Expert Systems with Applications*, 39(18):13390–13398, 2012.

David F Rogers, Robert D Plante, Richard T Wong, and James R Evans. Aggregation and disaggregation techniques and methodology in optimization. *Operations Research*, 39(4):553–582, 1991.

Anna Salleh. Aborigines may have farmed eels, built huts. *News in Science*, 2003.

Budi Santosa, Rita Damayanti, and Biswajit Sarkar. Solving multi-product inventory ship routing with a heterogeneous fleet model using a hybrid cross entropy-genetic algorithm: a case study in indonesia. *Production & Manufacturing Research*, 4(1):90–113, 2016.

Trygve Sigholt, Torbjørn Åsgård, and Magne Staurnes. Timing of parr-smolt transformation in atlantic salmon (salmo salar): effects of changes in temperature and photoperiod. *Aquaculture*, 160(1-2):129–144, 1998.

Oğuz Solyalı and Haldun Süral. A branch-and-cut algorithm using a strong formulation and an a priori tour-based heuristic for an inventory-routing problem. *Transportation Science*, 45(3):335–345, 2011.

Hugo Steinhaus. Sur la division des corp materiels en parties. *Bull. Acad. Polon. Sci*, 1 (804):801, 1956.

Éric D Taillard. A heuristic column generation method for the heterogeneous fleet vrp. *RAIRO-Operations Research*, 33(1):1–14, 1999.

P Toth and D Vigo. The vehicle routing problem (society for industrial and applied mathematics, philadelphia). Technical report, ISBN 0-89871-579-2, 2002.

Andres Weintraub, S Guitart, and V Kohn. Strategic planning in forest industries. *European journal of operational research*, 24(1):152–162, 1986.

Christopher Whelan, Greg Harrell, and Jin Wang. Understanding the k-medians problem. In *Proceedings of the International Conference on Scientific Computing (CSC)*, page 219. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2015.

Trine Ytrestøyl, Turid Synnøve Aas, and Torbjørn Einar Åsgård. Resource utilisation of norwegian salmon farming in 2012. 2014.

Yugang Yu, Haoxun Chen, and Feng Chu. A new model and hybrid approach for large scale inventory routing problems. *European Journal of Operational Research*, 189(3): 1022–1040, 2008.

Paul H Zipkin. Bounds on the effect of aggregating variables in linear programs. *Operations Research*, 28(2):403–418, 1980.

Appendices

A1 - Explanation of Attached Files

Below is an explanation of the different folders and files attached to the delivery of this thesis.

1. Input Files Folder

a) **RoutingData12zones.m** Input file for the production zones test case.
b) **RoutingData19natural.m** Input file for the natural selection test case.
c) **RoutingData24kmeans.m** Input file for the k-means test case.

2. Arc-Load Formulations Folder

a) **ArcLoad.m** Matrix implementation of standard arc-load formulation
b) **ArcLoadCliqueInequalities.m** Matrix implementation of arc-load formulation with clique inequalities added.
c) **ArcLoadInternal.m** Matrix implementation of arc-load formulation without external deliveries possible.
d) **ArcLoadLimitedCompatibility.m** Matrix implementation of standard arc-load formulation with limited ship harbour compatibility.
e) **ArcLoadSubTour.m** Matrix implementation of arc-load formulation with sub-tour elimination constraints added.
f) **ArcLoadSubTourAggregated.m** Matrix implementation of arc-load formulation with aggregated sub-tour elimination constraints added.
g) **ArcLoadTimeWindow.m** Matrix implementation of standard arc-load formulation.

3. Arc-Flow Formulations Folder

a) **ArcFlow.m** Matrix implementation of standard arc-flow formulation
b) **ArcFlowCliqueInequalities.m** Matrix implementation of arc-flow formulation with clique inequalities added.
c) **ArcFlowLimitedCompatibility.m** Matrix implementation of standard arc-flow formulation with limited ship harbour compatibility.
d) **ArcFlowSubTour.m** Matrix implementation of arc-flow formulation with sub-tour elimination constraints added.
e) **ArcFlowSubTourAggregated.m** Matrix implementation of arc-flow formulation with aggregated sub-tour elimination constraints added.
f) **ArcFlowTimeWindow.m** Matrix implementation of standard arc-flow formulation.

4. Decomposition Formulation Folder

a) **DantzigWolfeMaster.m** The master problem for the decomposition. Does not include the branch-and-bound part of the algorithm.
b) **DantzigWolfeHarbour.m** The harbour sub-problem problem for the decomposition.
c) **DantzigWolfeVessel.m** The vessel sub-problem problem for the decomposition.

5. Supporting Files

a) **BranchAndBound.m** Branch and Bound algorithm that can solve models on the same form as Gurobi input models.
b) **ResultMatrix.m** Plots the results in matrix and as graphs.

A2 - Complete Arc-Load Model

Definitions

Set	Definition
H_T	Set of all harbours
V	Set of all vessels
H_v	Set of all harbours, can be visited by vessel v
M_{Ti}	Set of possible arrivals at port i
M_{iv}	Set of possible arrivals at port i by ship v
A_v	Set of all possible arches for vessel v

Varables	Definition	Unit	Variable Type
x_{imjnv}	1 if ship v routed directly form node (i,m) to (j,n), 0 otherwise	[-]	Binary
y_{im}	1 if node (i,m) not visited by any ship, 0 otherwise	[-]	Binary
z_{imv}	1 if route ends at node (i,m) for ship v, 0 otherwise	[-]	Binary
u_i	1 if cluster served by external vessels, 0 otherwise	[-]	Binary
q_{imv}	Quantity loaded or unloaded at node (i,m) by ship v	tonnes	Continuous
l_{imv}	Quantity on ship v after visiting node (i,m)	tonnes	Continuous
t^S_{im}	Start time loading at node (i,m)	hours	Continuous
t^E_{im}	End time loading at node (i,m)	hours	Continuous
s^S_{im}	Stock level at start of loading at node (i,m)	tonnes	Continuous
s^E_{im}	Stock level at end of loading at node (i,m)	tonnes	Continuous

Parameters	Definition	Unit
C_{ijv}	Cost of sailing arc (i,j) with vessel v	USD
E	Extra cost for feed with external delivery	USD/tonn
E_i	Transportation cost for external deliveries to node i	USD
W_{imv}	1 if ship v starts at port i at arrival m, 0 otherwise	[-]
J_i	1 if i is a load harbour, -1 if discharge	[-]
Q^{MAX}_{imv}	Upper load limit	tonnes
Q^{MIN}_{im}	Lower load limit	tonnes
C^{AP}_v	Capacity vessel v	tonnes
T	Length of planning period	hours
T^Q_i	Time to unload one unit at port i	tonnes/hour
T^S_{ijv}	Time to sail from i to j with ship v	hours
T^B_i	Minimum time from departure till next arrival at port i	hours
T^{WS}_{im}	Start of time window for arrival m at port i	hours
T^{WE}_{im}	End of time window for arrival m at port i	hours
R_i	Production rate (Negative if consumption)	tonnes/hour
S^{MIN}_i	Minimum inventory level	tonnes
S^{EMIN}_i	Minimum inventory level at end of planning period	tonnes
S^{MAX}_i	Maximum inventory level	tonnes
S^S_i	Stock levels at node i at start	tonnes

Objective

$$minz = \sum_{v \in V} \sum_{A_v \in (i,m,j,n)} C_{ijv} x_{imjnv} + T \cdot E \sum_{i \in H_T} R_i(1-u_i) + \sum_{i \in H_T} E_i^T R_i(1-u_i) \quad \text{(A2.1)}$$

Network constraints

$$W_{imv} + \sum_{j \in H_v} \sum_{n \in M_{jv}} x_{jnimv} - \sum_{j \in H_v} \sum_{n \in M_{jv}} x_{imjnv} - z_{imv} = 0, \forall v \in V, i \in H_v, m \in M_{iv} \quad \text{(A2.2)}$$

$$\sum_{v \in V} \sum_{j \in H_v} \sum_{n \in M_{jv}} x_{jnimv} + y_{im} + \sum_{v \in V} W_{imv} = 1, \forall i \in H_T, m \in M_{Ti} \quad \text{(A2.3)}$$

$$y_{im} - y_{i(m-1)} \geq 0, \forall i \in H_T, m \in M_{Ti} \quad \text{(A2.4)}$$

Loading and unloading constraints

$$l_{imv} + J_j q_{jnv} - l_{jnv} + C_v^{AP} x_{imjnv} \leq C_v^{AP}, \forall v \in V, (i,m,j,n) \in A_v \quad \text{(A2.5)}$$

$$l_{imv} + J_j q_{jnv} - l_{jnv} - C_v^{AP} x_{imjnv} \geq -C_v^{AP}, \forall v \in V, (i,m,j,n) \in A_v \quad \text{(A2.6)}$$

$$l_{imv} - \sum_{j \in H_v} \sum_{n \in M_{jv}} C_v^{AP} x_{jnimv} \leq 0, \forall v \in V, i \in H_v, m \in M_{iv} \quad \text{(A2.7)}$$

$$q_{imv} - \sum_{j \in H_v} \sum_{n \in M_{jv}} Q_{imv}^{MAX} x_{jnimv} \leq W_{imv} Q_{imv}^{MAX}, \forall v \in V, i \in H_v, m \in M_{iv} \quad \text{(A2.8)}$$

$$\sum_{v \in V} q_{imv} + Q_{im}^{MIN} y_{im} \geq Q_{im}^{MIN}, \forall i \in H_T, m \in M_{Ti} \quad \text{(A2.9)}$$

Time constraints

$$t_{im}^S + \sum_{v \in V} T_i^Q q_{imv} - t_{im}^E = 0, \forall i \in H_T, m \in M_{Ti} \quad \text{(A2.10)}$$

$$(t_{im}^E + T_{ijv}^S - t_{jn}) + M x_{imjnv} \leq M, \forall v \in V m(i,m,j,n)\ in A_v \quad \text{(A2.11)}$$

$$t_{im}^S - t_{Ei(m-1)} + T_i^B y_{im} \geq T_i^B, \forall i \in H_T, m \in M_{Ti} \quad \text{(A2.12)}$$

$$T_{im}^{WS} \leq t_{im}^S \leq T_{im}^{WE}, \forall i \in H_T, m \in M_{Ti} \quad \text{(A2.13)}$$

Inventory constraints

$$s^S_{im} - \sum_{v \in V} J_i q_{imv} + R_i t^E_{im} - R_i t^S_{im} - s^E_{im} = 0, \forall i \in H_T, m \in M_{Ti} \tag{A2.14}$$

$$s_{Ei(m-1)} + R_i t^S_{im} - R_i t_{Ei(m-1)} - s^S_{im} = 0, \forall i \in H_T, m \in M_{Ti} \tag{A2.15}$$

$$S^{MIN}_i u_i \leq s^S_{im} \leq S^{MAX}_i, \forall i \in H_T, m \in M_{Ti} \tag{A2.16}$$

$$S^{MIN}_i u_i \leq s^E_{im} \leq S^{MAX}_i, \forall i \in H_T, m \in M_{Ti} \tag{A2.17}$$

$$S^{EMIN}_i u_i \leq s^E_{im} + R_i(T u_i - t^E_{im}) \leq S^{MAX}_i, \forall i \in H_T, m \in |M_{Ti}| \tag{A2.18}$$

$$S^S_i + R_i(t^E_{i1}) = s^S_{i1}, \forall i \in H_T \tag{A2.19}$$

Variable constraints

$$x_{imjnv} \in \{0,1\}, \forall v \in V, (i,m,j,n) \in A_v \tag{A2.20}$$

$$y_{imv} \in \{0,1\}, \forall v \in V, i \in H_v, m \in M_{iv} \tag{A2.21}$$

$$z_{imv} \in \{0,1\}, \forall v \in V, i \in H_v, m \in M_{iv} \tag{A2.22}$$

$$u_i \in \{0,1\}, \forall i \in H_T \tag{A2.23}$$

$$q_{imv} \geq 0, \forall v \in V, i \in H_v, m \in M_{iv} \tag{A2.24}$$

$$l_{imv} \geq 0, \forall v \in V, i \in H_v, m \in M_{iv} \tag{A2.25}$$

$$t^S_{im} \geq 0, \forall i \in H_T, m \in M_{Ti} \tag{A2.26}$$

$$t^E_{im} \geq 0, \forall i \in H_T, m \in M_{Ti} \tag{A2.27}$$

$$s^S_{im} \geq 0, \forall i \in H_T, m \in M_{Ti} \tag{A2.28}$$

$$s^E_{im} \geq 0, \forall i \in H_T, m \in M_{Ti} \tag{A2.29}$$

A3 - Complete Arc-Flow Model

Definitions

Set	Definition
H_T	Set of all harbours
V	Set of all vessels
H_v	Set of all harbours, can be visited by vessel v
M_{Ti}	Set of possible arrivals at port i
M_{iv}	Set of possible arrivals at port i by ship v
A_v	Set of all possible arches for vessel v

Varables	Definition	Unit	Variable Type
x_{imjnv}	1 if ship v routed directly form node (i,m) to (j,n), 0 otherwise	[-]	Binary
y_{im}	1 if node (i,m) not visited by any ship, 0 otherwise	[-]	Binary
z_{imv}	1 if route ends at node (i,m) for ship v, 0 otherwise	[-]	Binary
u_i	1 if cluster served by external vessels, 0 otherwise	[-]	Binary
q_{imv}	Quantity loaded or unloaded at node (i,m) by ship v	tonnes	Continuous
l_{imjnv}	Quantity on ship v when sailing from (i,m) to (j,n)	tonnes	Continuous
t^S_{im}	Start time loading at node (i,m)	hours	Continuous
t^E_{im}	End time loading at node (i,m)	hours	Continuous
s^S_{im}	Stock level at start of loading at node (i,m)	tonnes	Continuous
s^E_{im}	Stock level at end of loading at node (i,m)	tonnes	Continuous

Parameters	Definition	Unit
C_{ijv}	Cost of sailing arc (i,j) with vessel v	USD
E	Extra cost for feed with external delivery	USD/tonn
E_i	Transportation cost for external deliveries to node i	USD
W_{imv}	1 if ship v starts at port i at arrival m, 0 otherwise	[-]
J_i	1 if i is a load harbour, -1 if discharge	[-]
Q^{MAX}_{imv}	Upper load limit	tonnes
Q^{MIN}_{im}	Lower load limit	tonnes
C^{AP}_v	Capacity vessel v	tonnes
T	Length of planning period	hours
T^Q_i	Time to unload one unit at port i	tonnes/hour
T^S_{ijv}	Time to sail from i to j with ship v	hours
T^B_i	Minimum time from departure till next arrival at port i	hours
T^{WS}_{im}	Start of time window for arrival m at port i	hours
T^{WE}_{im}	End of time window for arrival m at port i	hours
R_i	Production rate (Negative if consumption)	tonnes/hour
S^{MIN}_i	Minimum inventory level	tonnes
S^{EMIN}_i	Minimum inventory level at end of planning period	tonnes
S^{MAX}_i	Maximum inventory level	tonnes
S^S_i	Stock levels at node i at start	tonnes

Objective

$$minz = \sum_{v \in V} \sum_{A_v \in (i,m,j,n)} C_{ijv} x_{imjnv} + T \cdot E \sum_{i \in H_T} R_i(1-u_i) + \sum_{i \in H_T} E_i^T R_i (1-u_i) \quad \text{(A3.1)}$$

Network constraints

$$W_{imv} + \sum_{j \in H_v} \sum_{n \in M_{jv}} x_{jnimv} - \sum_{j \in H_v} \sum_{n \in M_{jv}} x_{imjnv} - z_{imv} = 0, \forall v \in V, i \in H_v, m \in M_{iv} \quad \text{(A3.2)}$$

$$\sum_{v \in V} \sum_{j \in H_v} \sum_{n \in M_{jv}} x_{jnimv} + y_{im} + \sum_{v \in V} W_{imv} = 1, \forall i \in H_T, m \in M_{Ti} \quad \text{(A3.3)}$$

$$y_{im} - y_{i(m-1)} \geq 0, \forall i \in H_T, m \in M_{Ti} \quad \text{(A3.4)}$$

Loading and unloading constraints

$$\sum_{j \in H_v} \sum_{n \in M_{jv}} l_{jnimv} + J_i q_{imv} - \sum_{j \in H_v} \sum_{n \in M_{jv}} l_{imjnv} = 0, \forall v \in V, i \in H_v, m \in M_{iv} \quad \text{(A3.5)}$$

$$l_{imjnv} \leq C_v^{AP} x_{imjnv}, \forall (i,m,j,n) \in A_v, v \in V \quad \text{(A3.6)}$$

$$q_{imv} - \sum_{j \in H_v} \sum_{n \in M_{jv}} Q_{imv}^{MAX} x_{jnimv} \leq W_{imv} Q_{imv}^{MAX}, \forall v \in V, i \in H_v, m \in M_{iv} \quad \text{(A3.7)}$$

$$\sum_{v \in V} q_{imv} + Q_{im}^{MIN} y_{im} \geq Q_{im}^{MIN}, \forall i \in H_T, m \in M_{Ti} \quad \text{(A3.8)}$$

Time constraints

$$t_{im}^S + \sum_{v \in V} T_i^Q q_{imv} - t_{im}^E = 0, \forall i \in H_T, m \in M_{Ti} \quad \text{(A3.9)}$$

$$(t_{im}^E + T_{ijv}^S - t_{jn}) + M x_{imjnv} \leq M, \forall v \in V m(i,m,j,n)\ inA_v \quad \text{(A3.10)}$$

$$t_{im}^S - t_{Ei(m-1)} + T_i^B y_{im} \geq T_i^B, \forall i \in H_T, m \in M_{Ti} \quad \text{(A3.11)}$$

$$T_{im}^{WS} \leq t_{im}^S \leq T_{im}^{WE}, \forall i \in H_T, m \in M_{Ti} \quad \text{(A3.12)}$$

Inventory constraints

$$s_{im}^{S} - \sum_{v \in V} J_i q_{imv} + R_i t_{im}^{E} - R_i t_{im}^{S} - s_{im}^{E} = 0, \forall i \in H_T, m \in M_{Ti} \tag{A3.13}$$

$$s_{Ei(m-1)} + R_i t_{im}^{S} - R_i t_{Ei(m-1)} - s_{im}^{S} = 0, \forall i \in H_T, m \in M_{Ti} \tag{A3.14}$$

$$S_i^{MIN} u_i \leq s_{im}^{S} \leq S_i^{MAX}, \forall i \in H_T, m \in M_{Ti} \tag{A3.15}$$

$$S_i^{MIN} u_i \leq s_{im}^{E} \leq S_i^{MAX}, \forall i \in H_T, m \in M_{Ti} \tag{A3.16}$$

$$S_i^{EMIN} u_i \leq s_{im}^{E} + R_i(T u_i - t_{im}^{E}) \leq S_i^{MAX}, \forall i \in H_T, m \in |M_{Ti}| \tag{A3.17}$$

$$S_i^{S} + R_i(t_{i1}^{E}) = s_{i1}^{S}, \forall i \in H_T \tag{A3.18}$$

Variable constraints

$$x_{imjnv} \in \{0,1\}, \forall v \in V, (i,m,j,n) \in A_v \tag{A3.19}$$

$$y_{imv} \in \{0,1\}, \forall v \in V, i \in H_v, m \in M_{iv} \tag{A3.20}$$

$$z_{imv} \in \{0,1\}, \forall v \in V, i \in H_v, m \in M_{iv} \tag{A3.21}$$

$$u_i \in \{0,1\}, \forall i \in H_T \tag{A3.22}$$

$$q_{imv} \geq 0, \forall v \in V, i \in H_v, m \in M_{iv} \tag{A3.23}$$

$$l_{imjnv} \geq 0, \forall (i,m,j,n) \in A_v, v \in V \tag{A3.24}$$

$$t_{im}^{S} \geq 0, \forall i \in H_T, m \in M_{Ti} \tag{A3.25}$$

$$t_{im}^{E} \geq 0, \forall i \in H_T, m \in M_{Ti} \tag{A3.26}$$

$$s_{im}^{S} \geq 0, \forall i \in H_T, m \in M_{Ti} \tag{A3.27}$$

$$s_{im}^{E} \geq 0, \forall i \in H_T, m \in M_{Ti} \tag{A3.28}$$

A4 - Comprehensive Optimization Model

Mathematical model formulation

Set	Definition
H_T	Set of all ports
H_{Tv}	Set of all unloading ports
N	Set of all unloading ports for all visits
U	Set of loading port for all visits
A_v	Set of arcs possible for ship (v)
M_i	Set of possible arrivals at port i

Parameter	Definition	Unit
C^t_{ijv}	Cost of transport from node (i) to (j) by ship (v)	USD
C^p_{iv}	Setup cost in port (i) for ship (v)	USD
C^{rv}_v	Cost of renting ship (v)	USD
C^c_{vc}	Cost of cleaning compartment (c) in ship (v)	USD
A^v_v	Capacity vessel v	tonnes
A^i_i	Unloading capacity port (i)	tonnes
TC_{vc}	Time to clean compartment c in ship v	hours
J_{ik}	1 if product k being loaded or unloaded in port (i)	[-]
QS_{ik}	Shipping quantity of product (k) in port (i)	tonnes
PH	Planning horizon	hours
TP	Port setup time	hours
TQu_{ik}	Unloading speed product (k) in port (i)	tonnes/hour
TQl_{ik}	Loading speed product (k) in port (i)	tonnes/hour
TT_{ijv}	Travelling time between node (i) and (j) for vessel (v)	hours
$[A_i, B_i]$	Time window port (i)	hours
H_{ik}	Inventory at start for product (k) at port (i)	tonnes
$Smin_{ik}$	Minimum inventory of product (k) at port (i)	tonnes
$Smin^E_{ik}$	Minimum inventory of product (k) at port (i) at EOH	tonnes
$Smax_{ik}$	Maximum inventory of product (k) at port (i)	tonnes
CR_{ik}	Consumption rate of product (k) at node (i)	tonnes/hour

Variables	Definition	Unit
x_{imjnv}	1 if ship moves from unloading port (i) after visit (m) to unloading port (j) at visit (n) for ship (v)	[-]
x_{imdnv}	1 if ship moves from unloading port (i) after visit (m) to loading port (d) at visit (n) for ship (v)	[-]
y_{im}	1 if port (i) is not reachable for visit (m)	[-]
r_{ik}	1 if the stock level of product (k) at unloading port (i) is able to meet consumption needs	[-]
u_{iv}	1 if capacity of ship (v) is smaller than capacity of unloading port (i)	[-]
z_{vcm}	1 if compartment (c) of ship (v) is washed at visit (m)	[-]
by_{dmkvc}	1 if there is product (k) in compartment (c) at ship (v) that loaded in port (d) at visit (m)	[-]
xy_{imkv}	1 if product (k) is brought by ship (v) at visit (m) at port (i)	[-]
I_{imvkc}	Amount of product (k) in compartment (c) that ship (v) carries after leaving on visit (m) at port (i)	tonnes
q_{imvkc}	Amount of product (k) unloaded from compartment (c) in ship (v) at visit (m) in port (i)	tonnes
q_{imvkc}	Amount of product (k) loaded from compartment (c) in ship (v) at visit (m) in port (d)	tonnes
p_imv	1 if ship (v) visits port (i) on visit (m)	[-]
p_imv	1 if ship (v) visits loading port (d) on visit (m)	[-]
ta_{im}	Arrival time in port (i) on visit (m)	hours
te_{im}	Ending service time in port (i) on visit (m)	hours
s_{imk}	Amount of product (k) in port (i) before unloading at visit (m)	tonnes

Objective

$$\sum_{v\in V}\sum_{(i,m,j,n)\in A_v} c_{ij}x_{imjnv} + \sum_{v\in V}\sum_{(d,m,j,n)\in A_v} c_{djv}(x_{dmjnv}+x_{imdnv}) + \sum_{v\in V} cp_{iv}(\sum_{(i,m)\in N} p_{imv} + \sum_{(d,m)\in N} p_{dmv}) + \sum_{v\in V} crv_v \sum_{(d,m)\in U}(ta_{dm} - ta_{d(m-1)}) + \sum_{v\in V}\sum_{c\in C_v} cc_{vc}z_{vc} \quad \text{(A4.1)}$$

Constrains

$$\sum_{(d,m)\in U}\sum_{(j,n)\in N} x_{jndmv} - \sum_{(d,m)\in U}\sum_{(j,n)\in N} x_{jndmv} * u_{iv} - r_{ik} = 0, u_{iv} = \begin{cases} 1, A_v^v \le A_i^i \\ 0 otherwise \end{cases} \quad \text{(A4.2)}$$

$$r_{ik} = \begin{cases} 1, s_{imk} + q_{imvkc} \ge CR_{ik}(PH - te_{im}) \\ 0, s_{imk} + q_{imvkc} \le CR_{ik}(PH - te_{im}) \end{cases} \quad ,\forall v \in V, \forall k \in K, \forall (v,d,m) \in VxU, \forall (v,i,m \quad \text{(A4.3)}$$

$$\sum\sum x_{jnimv} + y_{im} = 1, \forall (i,m) \in N, i \ne j \quad \text{(A4.4)}$$

$$y_{im} - y_{i(m-1)} \ge 0, \forall (i,m) \in N, m \ne 1 \quad \text{(A4.5)}$$

$$y_{dm} - y_{d(m-1)} \ge 0, \forall (d,m) \in U, m \ne 1 \quad \text{(A4.6)}$$

$$x_{dmjnv} \in \{0,1\} \forall v \in V, \forall (d,m,j,n) \in A_v \quad \text{(A4.7)}$$

$$x_{imjnv} \in \{0,1\} \forall v \in V, \forall (i,m,j,n) \in A_v \quad \text{(A4.8)}$$

$$u_{iv}, r_{ik} \in \{0,1\} \forall k \in K, \forall v \in V, \forall i \in H_{Tv} \quad \text{(A4.9)}$$

$$y_{im} \in \{0,1\} \forall (i,m) \in N \quad \text{(A4.10)}$$

$$z_{vc} = \begin{cases} 1, by_{dmkvc} - by_{d(m+1)k_v vc} = 0 \\ 0, by_{dmkvc} - by_{d(m+1)k_v vc} \ne 0 \end{cases} \quad \forall v \in V, \forall (d,m) \in U, \forall k_v \in K_v, \forall (k,c) \in K_v x C_v, k_v \ne k \quad \text{(A4.11)}$$

$$\sum_{k\in K} by_{dmvck} \le 1, \forall v \in V, \forall (d,m) \in U, \forall (k,c) \in K_v x C_v \quad \text{(A4.12)}$$

$$\sum_{c\in C_v} by_{dmvck} \leq |C_v| \cdot (1 - by_{dmvc_c k_k}), \forall v \in V, \forall k \in K, \forall (d,m) \in U, \forall c_c \in C_v, \forall k_k \in K_v, \forall (k,c) \in K_v \tag{A4.13}$$

$$z_{vc} \in \{0,1\}, \forall v \in V, \forall c \in C_v \tag{A4.14}$$

$$by_{dmkvc} \in \{0,1\}, \forall v \in V, \forall (d,m) \in U, \forall (k,c) \in K_v x C_v \tag{A4.15}$$

$$\sum_{v\in V} xy_{imkv} = 1, \forall k \in K_v, \forall (i,m)\ inN \tag{A4.16}$$

$$x_{imjnv}(I_{imvkc} - J_{ik} q_{jnvkc} - I_{jnvkc}) = 0, \forall v \in V, \forall (i,m,j,n) \in N, \forall (k,c) \in K_v x C_v, i \neq j \tag{A4.17}$$

$$x_{dmjnv}(J_{ik} q_{dmvkc} - I_{jnvkc}) = 0, \forall v \in V, \forall (d,m) \in U, \forall (j,n) \in N \forall (k,c) \in K_v x C_v, j \neq d \tag{A4.18}$$

$$\sum_{c\in C} \sum_{k\in K_v} q_{dmvkc} = \sum_{(i,m)\in N} \sum_{k\in K_v} xy_{imkv} QS_{ik} + (\sum_{c\in C} CMax_{vc} - \sum_{(i,m)\in N} \sum_{k\in K_v} xy_{imkc} QS_{ik}), \forall v \in V, \forall (d,m) \in U \tag{A4.19}$$

$$q_{dmvkc} \leq CMac_{vc}, \forall v \in V, \forall (d,m) \in U, \forall (k,c) \in K_v x C_c, i \neq d \tag{A4.20}$$

$$q_{dmvkc} \leq CMac_{vc} \sum_{(j,n)\in N} x_{jnimv}, \forall v \in V, \forall (i,m) \in N, \forall (k,c) \in K_v x C_c, i \neq j \tag{A4.21}$$

$$I_{dmvkc} \leq CMac_{vc} \sum_{(j,n)\in N} x_{jnimv}, \forall v \in V, \forall (i,m) \in N, \forall (k,c) \in K_v x C_c, i \neq j \tag{A4.22}$$

$$x_{jnimv} - p_{imv} = 0, \forall v \in V, \forall (i,m,j,n) \in N \tag{A4.23}$$

$$x_{jnimv} - p_{dmv} = 0, \forall v \in V, \forall (j,n) \in N, \forall (d,m) \in U \tag{A4.24}$$

$$xy_{imvk} \in \{0,1\}, \forall v \in V, \forall k \in K, \forall (i,m) \in N \tag{A4.25}$$

$$p_{imv}, p_{dmv} \in \{0,1\}, \forall v \in V, \forall (i,m) \in N, \forall (d,m) \in U \tag{A4.26}$$

$$I_{imvkc}, q_{imvkc}, q_{dmvkc} \geq 0, \forall v \in V, \forall (i,m) \in N, \forall (d,m) \in U, \forall (k,c) \in K_v x C_v \quad \text{(A4.27)}$$

$$ta_{i(m+1)} - ta_{im} \geq 0, \forall (i,m) \in N \quad \text{(A4.28)}$$

$$ta_{d(m+1)} - ta_{dm} \geq 0, \forall (d,m) \in U \quad \text{(A4.29)}$$

$$ta_{im} + TP_i p_{imv} \leq [\frac{H_{ik} - SMin_{ik}}{CR_{ik}}], \forall k \in K, \forall (i,m) \in N \quad \text{(A4.30)}$$

$$A_i \leq ta_{im} \leq B_i, \forall (i,m) \in N \quad \text{(A4.31)}$$

$$ta_{im} + TP_i p_{imv} + \sum_{k \in K_v} \sum_{c \in C} TQi_{ik} q_{imvkc} - te_{im} = 0, \forall v \in V, \forall (i,m) \in N, \forall (k,c) \in K_v x C_v \quad \text{(A4.32)}$$

$$ta_{dm} + TP_i p_{dmv} + \sum_{k \in K_v} \sum_{c \in C} TQl_{dk} q_{dmvkc} + \sum_{c \in C} TC_{vc} z_{vc} - te_{dm} = 0, \forall v \in V, \forall (d,m) \in U \quad \text{(A4.33)}$$

$$x_{imjnv}[te_{im} + TT_{ijv} - ta_{jn}] \leq 0, \forall v \in V, \forall (i,m,j,n) \in A_v \quad \text{(A4.34)}$$

$$x_{dmjnv}[te_{dm} + TT_{djv} - ta_{jn}] \leq 0, \forall v \in V, \forall (d,m,j,n) \in A_v \quad \text{(A4.35)}$$

$$x_{imdnv}[te_{im} + TT_{idv} - ta_{dn}] \leq 0, \forall v \in V, \forall (i,m,d,n) \in A_v \quad \text{(A4.36)}$$

$$ta_{im}, te_{im} \geq 0, \forall (i,m) \in N \quad \text{(A4.37)}$$

$$ta_{dm}, te_{dm} \geq 0, \forall (d,m) \in U \quad \text{(A4.38)}$$

$$s_{imk} = H_{ik} - CR_{ik}(ta_{im} + TP_i p_{imv}) \quad \text{(A4.39)}$$

$$SMin_{ik} \leq s_{imk} \leq SMax_{ik}, \forall (i,m,k) \in (H_T - d) x K_i \quad \text{(A4.40)}$$

$$SMin_{ik} \leq s_{imk} + q_{imkc} - CR_{ik}(te_{im} - (ta_{im} + TP_i p_{imv})) \leq SMax_{ik}, \forall v \in V, \forall c \in c_v, \forall (i,m,k) \in (H_T - d) x K_i \quad \text{(A4.41)}$$

$$SMin_{ik}^{E} \leq s_{imk} + q_{imkc} - CR_{ik}(te_{im} - (ta_{im} + TP_i p_{imv})) \leq SMax_{ik}, \forall v \in V, \forall c \in c_v, \forall (i,m,k) \in (H_T - d) x M_i \quad \text{(A4.42)}$$

A5 - MATLAB Matrix Implementation and Gurobi Model for Arc-Load Problem

Script starts on next page.

```
%Arc-Load Model

%Defines the problem
%    min z=C'x
%st. Ax<=B

%Clears all variables
clc,clear

%Loads routing, ship, harbour and consumption data
%RoutingData12zones()
%RoutingData19natural();
RoutingData24kmeans()

%Defines Variables--------------------------------------

%i,m,j,n,v
x=zeros(n_ht,n_mti,n_ht,n_mti,n_v);
[xa xb xc xd xe] = size(x);
xlength=xa*xb*xc*xd*xe;

%i,m
y=zeros(n_ht,n_mti);
[ya yb] = size(y);
ylength=ya*yb;

%i,m,v
z=zeros(n_ht,n_mti,n_v);
[za zb zc] = size(z);
zlength=za*zb*zc;

%i
u=zeros(n_ht,1);
[ua]=size(u);
ulength=ua;

%i,m,v
q=zeros(n_ht,n_mti,n_v);
[qa qb qc] = size(q);
qlength=qa*qb*qc;

%i,m,v
l=zeros(n_ht,n_mti,n_v);
[la lb lc] = size(l);
llength=la*lb*lc;

%i,m
ts=zeros(n_ht,n_mti);
[tsa tsb] = size(ts);
tslength=tsa*tsb;

%i,m
```

```
te=zeros(n_ht,n_mti);
[tea teb] = size(te);
telength=tea*teb;

%i,m
ss=zeros(n_ht,n_mti);
[ssa ssb] = size(ss);
sslength=ssa*ssb;

%i,m
se=zeros(n_ht,n_mti);
[sea seb] = size(se);
selength=sea*seb;


%Generate Index Numbers for Variables------------------

t1=0;


for i=1:n_ht
    for m=1:n_mt(i)
        for j=1:n_ht
            for n=1:n_mt(j)
                for v=1:n_v
                    t1=t1+1;
                    x(i,m,j,n,v)=t1;
                end
            end
        end
    end
end

xlength=t1;

for i=1:n_ht
    for m=1:n_mt(i)
        t1=t1+1;
        y(i,m)=t1;
    end
end

ylength=t1-xlength;

for i=1:n_ht
    for m=1:n_mt(i)
        for v=1:zc
            t1=t1+1;
            z(i,m,v)=t1;
        end
    end
end

zlength=t1-xlength-ylength;
```

```
for i=1:ua
            t1=t1+1;
            u(i)=t1;
end

ulength=t1-xlength-ylength-zlength;

for i=1:n_ht
    for m=1:n_mt(i)
        for v=1:qc
            t1=t1+1;
            q(i,m,v)=t1;
        end
    end
end

qlength=t1-xlength-ylength-zlength-ulength;

for i=1:n_ht
    for m=1:n_mt(i)
        for v=1:lc
            t1=t1+1;
            l(i,m,v)=t1;
        end
    end
end

llength=t1-xlength-ylength-zlength-ulength-qlength;

for i=1:tsa
    for m=1:tsb
        t1=t1+1;
        ts(i,m)=t1;
    end
end

tslength=t1-xlength-ylength-zlength-ulength-qlength-llength;

for i=1:tea
    for m=1:teb
        t1=t1+1;
        te(i,m)=t1;
    end
end

telength=t1-xlength-ylength-zlength-ulength-qlength-llength-tslength;

for i=1:ssa
    for m=1:ssb
        t1=t1+1;
        ss(i,m)=t1;
    end
end
```

```
sslength=t1-xlength-ylength-zlength-ulength-qlength-llength-tslength-
telength;

for i=1:sea
    for m=1:seb
        t1=t1+1;
        se(i,m)=t1;
    end
end

selength=t1-xlength-ylength-zlength-ulength-qlength-llength-tslength-
telength-sslength;

%Generate C vector
C=zeros(1, xlength+ylength+zlength+ulength+qlength+llength+telength
+tslength+sslength+selength);

for v=1:n_v
    for i=1:n_ht
        for m=1:n_mt(i)
            for j=1:n_ht
                for n=1:n_mt(j)
                    index1=x(i,m,j,n,v);
                    C(index1)=C(index1)+CS(i,j,v);
                end
            end
        end
    end
end


for i=2:n_ht
    index1=u(i);
    C(index1)=C(index1)-T*EF*(-1)*R(i)-ET(i);
end

C_const=0;
for i=2:n_ht
    C_const=C_const+T*EF*(-1)*R(i)+ET(i);
end

%Generate A matrix-------------------------------

A=sparse(100000,xlength+ylength+zlength+ulength+qlength+llength
+telength+tslength+sslength+selength);

%Set Constraints---------------------------------

%constraint 1
t1=0;
for v=1:n_v
    for i=1:n_ht
        for m=1:n_mt(i)
```

```
            t1=t1+1;
            model.sense(t1)='=';
            B(t1)=-W(i,m,v);
            index3=z(i,m,v);
            for j=1:n_ht
                for n=1:n_mt(j)

            index1=x(j,n,i,m,v);
            index2=x(i,m,j,n,v);

            A(t1,index1)=A(t1,index1)+1;
            A(t1,index2)=A(t1,index2)-1;

                end
            end
            A(t1,index3)=A(t1,index3)-1;
        end
    end
end



%constraint 2

t2=t1;
for i=1:n_ht
    for m=1:n_mt(i)
        t2=t2+1;
        model.sense(t2)='=';
        B(t2)=1-sum(W(i,m,:));

        index2=y(i,m);
        A(t2,index2)=A(t2,index2)+1;

        for j=1:n_ht
            for n=1:n_mt(j)
                for v=1:n_v
            index1=x(j,n,i,m,v);
            A(t2,index1)=1;
                end
            end
        end
    end
end

%constraint 3

t3=t2;
for i=1:n_ht
    for m=2:n_mt(i)
        t3=t3+1;
        model.sense(t3)='>';
        B(t3)=0;
        index1=y(i,m);
        index2=y(i,m-1);

        A(t3,index1)=A(t3,index1)+1;
        A(t3,index2)=A(t3,index2)-1;
    end
end

%costraint 4
t4=t3;
for v=1:n_v
    for i=1:n_ht
        for m=1:n_mt(i)
            for j=1:n_ht
                for n=1:n_mt(j)
                    t4=t4+1;
                    model.sense(t4)='<';
                    B(t4)=CAP(v);
                    index1=x(i,m,j,n,v);
                    index2=l(i,m,v);
                    index3=q(j,n,v);
                    index4=l(j,n,v);
                    A(t4,index1)=A(t4,index1)+CAP(v);
                    A(t4,index2)=A(t4,index2)+1;
                    A(t4,index3)=A(t4,index3)+J(j);
                    A(t4,index4)=A(t4,index4)-1;
                end
            end
        end
    end
end

%costraint 5
t5=t4;
for v=1:n_v
    for i=1:n_ht
        for m=1:n_mt(i)
            for j=1:n_ht
                for n=1:n_mt(j)
                    t5=t5+1;
                    model.sense(t5)='>';
                    B(t5)=-CAP(v);
                    index1=x(i,m,j,n,v);
                    index2=l(i,m,v);
                    index3=q(j,n,v);
                    index4=l(j,n,v);
                    A(t5,index1)=A(t5,index1)-CAP(v);
                    A(t5,index2)=A(t5,index2)+1;
                    A(t5,index3)=A(t5,index3)+J(j);
                    A(t5,index4)=A(t5,index4)-1;
                end
            end
        end
    end
end
```

```
%constraint 6
t6=t5;
for v=1:n_v
    for i=1:n_ht
        for m=1:n_mt(i)
            t6=t6+1;
            model.sense(t6)='<';
            B(t6)=0;
            index1=l(i,m,v);
            A(t6,index1)=A(t6,index1)+1;
            for j=1:n_ht
                for n=1:n_mt(j)
            index2=x(i,m,j,n,v);
            A(t6,index2)=A(t6,index2)-CAP(v);
                end
            end
        end
    end
end

%constraint 7
t7=t6;
for v=1:n_v
    for i=1:n_ht
        for m=1:n_mt(i)
            t7=t7+1;
            model.sense(t7)='<';
            B(t7)=W(i,m,v)*QMax(i,m,v);
            index1=q(i,m,v);
            A(t7,index1)=A(t7,index1)+1;
            for j=1:n_ht
                for n=1:n_mt(j)
            index2=x(j,n,i,m,v);
            A(t7,index2)=A(t7,index2)-QMax(i,m,v);%change
 --------------------
                end
            end
        end
    end
end

%constraint 8
t8=t7;
for i=1:n_ht
    for m=1:n_mt(i)
        t8=t8+1;
        model.sense(t8)='>';
        B(t8)=QMin(i,m);
        index1=y(i,m);
        A(t8,index1)=A(t8,index1)+QMin(i,m);
        for v=1:n_v
        index2=q(i,m,v);
        A(t8,index2)=A(t8,index2)+1;
        end
    end
end

%Time Constraints----------------------------

%constraint 9
t9=t8;
for i=1:n_ht
    for m=1:n_mt(i)
        t9=t9+1;
        model.sense(t9)='=';
        B(t9)=0;
        index1=ts(i,m);
        index2=te(i,m);
        A(t9,index1)=A(t9,index1)+1;
        A(t9,index2)=A(t9,index2)-1;
        for v=1:n_v
        index3=q(i,m,v);
        A(t9,index3)=A(t9,index3)+TQ(i);
        end
        for v=1:n_v
            for j=1:n_ht
                for n=1:n_mt(j)
                index4=x(i,m,j,n,v);
                A(t9,index4)=A(t9,index4)+TSI(i,1);
                end
            end
        end
    end
end

%constraint 10
t10=t9;
for v=1:n_v
    for i=1:n_ht
        for m=1:n_mt(i)
            for j=1:n_ht
                for n=1:n_mt(j)
                    t10=t10+1;
                    model.sense(t10)='<';
                    B(t10)=M-TS(i,j,v);
                    index1=te(i,m);
                    index2=ts(j,n);
                    index3=x(i,m,j,n,v);
                    A(t10,index1)=A(t10,index1)+1;
                    A(t10,index2)=A(t10,index2)-1;
                    A(t10,index3)=A(t10,index3)+M;
                end
            end
        end
    end
end
```

```
%constraint 11
%This constraint was eliminated
t11=t10;

%constraint 12
t12=t11;
for i=1:n_ht
    for m=2:n_mt(i)
        t12=t12+1;
        model.sense(t12)='>';
        B(t12)=TB(i);
        index1=ts(i,m);
        index2=te(i,m-1);
        index3=y(i,m);
        A(t12,index1)=A(t12,index1)+1;
        A(t12,index2)=A(t12,index2)-1;
        A(t12,index3)=A(t12,index3)+TB(i);
    end
end

%constraint 13
t13=t12;
for i=1:n_ht
    for m=1:n_mt(i)
        t13=t13+1;
        model.sense(t13)='>';
        B(t13)=TWS(i,m);
        index1=ts(i,m);
        A(t13,index1)=A(t13,index1)+1;
    end
end

%constraint 14
t14=t13;
for i=1:n_ht
    for m=1:n_mt(i)
        t14=t14+1;
        model.sense(t14)='<';
        B(t14)=TWE(i,m);
        index1=ts(i,m);
        A(t14,index1)=A(t14,index1)+1;
    end
end

%Inventory Constraints-----------------------
%constraint 15
t15=t14;
for i=1:n_ht
    for m=1:n_mt(i)
        t15=t15+1;
        model.sense(t15)='=';
        B(t15)=0;
        index1=ss(i,m);
```

```
            index3=te(i,m);
            index4=ts(i,m);
            index5=se(i,m);
            A(t15,index1)=A(t15,index1)+1;
            for v=1:n_v
                index2=q(i,m,v);
                A(t15,index2)=A(t15,index2)-J(i);
            end
            A(t15,index3)=A(t15,index3)+R(i);
            A(t15,index4)=A(t15,index4)-R(i);
            A(t15,index5)=A(t15,index5)-1;
        end
    end

    %constraint 16
    t16=t15;
    for i=1:n_ht
        for m=2:n_mt(i)
            t16=t16+1;
            model.sense(t16)='=';
            B(t16)=0;
            index1=ss(i,m);
            index3=te(i,m-1);
            index4=ts(i,m);
            index5=se(i,m-1);
            A(t16,index1)=A(t16,index1)-1;
            A(t16,index3)=A(t16,index3)-R(i);
            A(t16,index4)=A(t16,index4)+R(i);
            A(t16,index5)=A(t16,index5)+1;
        end
    end

    %constraint 17
    t17=t16;
    for i=1:n_ht
        for m=1:n_mt(i)
            t17=t17+1;
            model.sense(t17)='>';
            B(t17)=SMin(i);
            index1=ss(i,m);
            A(t17,index1)=A(t17,index1)+1;
        end
    end

    %constraint 18
    t18=t17;
    for i=1:n_ht
        for m=1:n_mt(i)
            t18=t18+1;
            model.sense(t18)='<';
            B(t18)=SMax(i);
            index1=ss(i,m);
            A(t18,index1)=A(t18,index1)+1;
        end
```

```
end

%constraint 19
t19=t18;
for i=1:n_ht
    for m=1:n_mt(i)
        t19=t19+1;
        model.sense(t19)='>';
        B(t19)=SMin(i);
        index1=se(i,m);
        A(t19,index1)=A(t19,index1)+1;
    end
end

%constraint 20
t20=t19;
for i=1:n_ht
    for m=1:n_mt(i)
        t20=t20+1;
        model.sense(t20)='<';
        B(t20)=SMax(i);
        index1=se(i,m);
        A(t20,index1)=A(t20,index1)+1;
    end
end

%constraint 21
t21=t20;
for i=1:n_ht
    m=n_mt(i);
        t21=t21+1;
        model.sense(t21)='>';
        B(t21)=0;
        index1=se(i,m);
        index2=te(i,m);
        index3=u(i);
        A(t21,index1)=A(t21,index1)+1;
        A(t21,index2)=A(t21,index2)-R(i);
        A(t21,index3)=A(t21,index3)-SEMin(i)+R(i)*T;
end

%constraint 22
t22=t21;
for i=1:n_ht
    m=n_mt(i);
        t22=t22+1;
        model.sense(t22)='<';
        B(t22)=SMax(i)-R(i)*T;
        index1=se(i,m);
        index2=te(i,m);
        A(t22,index1)=A(t22,index1)+1;
        A(t22,index2)=A(t22,index2)-R(i);
end
```

```
%constraint 23
t23=t22;
for i=1:n_ht
    m=1;
        t23=t23+1;
        model.sense(t23)='=';
        B(t23)=SS(i);
        index1=ss(i,m);
        index2=ts(i,m);
        A(t23,index1)=A(t23,index1)+1;
        A(t23,index2)=A(t23,index2)-R(i);
end

%assignes values to the Gurobi model
model.vtype(1:xlength+ylength+zlength+ulength)='B';%defines binary
 variables
model.vtype(xlength+ylength+zlength+ulength+1:xlength+ylength
+zlength+ulength+qlength+llength+telength+tslength+sslength
+selength)='C';%defines continuous variables

%A( all(~A,2), : ) = []; %removes rows with zeros
AN=sparse(t23,xlength+ylength+zlength+ulength+qlength+llength+telength
+tslength+sslength+selength);
AN=A(1:t23,1:xlength+ylength+zlength+ulength+qlength+llength+telength
+tslength+sslength+selength);

model.A=sparse(AN);
model.rhs=B;
model.obj=C;
model.objcon=C_const;
model.modelsense='Min';

%Solves the problem in Gurobi
result= gurobi(model)
a=result.x
```

Published with MATLAB® R2018b

A6 - MATLAB Matrix Implementation and Gurobi Model for Arc-Flow Problem

Script starts on next page.

```
%Arc-Flow Model

%Defines the problem
%    min z=C'x
%st. Ax<=B

clc,clear

%Loads routing, ship, harbour and consumption data
%RoutingData12zones()
%RoutingData19natural()
RoutingData24kmeans()

modelType='Arc-Flow';

%Defines Variables--------------------------------------

%i,m,j,n,v
x=zeros(n_ht,n_mti,n_ht,n_mti,n_v);
[xa xb xc xd xe] = size(x);
xlength=xa*xb*xc*xd*xe;

%i,m
y=zeros(n_ht,n_mti);
[ya yb] = size(y);
ylength=ya*yb;

%i,m,v
z=zeros(n_ht,n_mti,n_v);
[za zb zc] = size(z);
zlength=za*zb*zc;

%i
u=zeros(n_ht,1);
[ua]=size(u);
ulength=ua;

%i,m,v
q=zeros(n_ht,n_mti,n_v);
[qa qb qc] = size(q);
qlength=qa*qb*qc;

%i,m,k,n,v
l=zeros(n_ht,n_mti,n_ht,n_mti,n_v);
[la lb lc ld le] = size(l);
llength=la*lb*lc*ld*le;

%i,m
ts=zeros(n_ht,n_mti);
[tsa tsb] = size(ts);
tslength=tsa*tsb;
```

```
%i,m
te=zeros(n_ht,n_mti);
[tea teb] = size(te);
telength=tea*teb;

%i,m
ss=zeros(n_ht,n_mti);
[ssa ssb] = size(ss);
sslength=ssa*ssb;

%i,m
se=zeros(n_ht,n_mti);
[sea seb] = size(se);
selength=sea*seb;


%Generate Index Numbers for Variables------------------

t1=0;


for i=1:n_ht
    for m=1:n_mt(i)
        for j=1:n_ht
            for n=1:n_mt(j)
                for v=1:n_v
                    t1=t1+1;
                    x(i,m,j,n,v)=t1;
                end
            end
        end
    end
end

xlength=t1;

for i=1:n_ht
    for m=1:n_mt(i)
        t1=t1+1;
        y(i,m)=t1;
    end
end

ylength=t1-xlength;

for i=1:n_ht
    for m=1:n_mt(i)
        for v=1:zc
            t1=t1+1;
            z(i,m,v)=t1;
        end
    end
end
```

```
zlength=t1-xlength-ylength;

for i=1:ua
            t1=t1+1;
            u(i)=t1;
end

ulength=t1-xlength-ylength-zlength;

for i=1:n_ht
    for m=1:n_mt(i)
        for v=1:qc
            t1=t1+1;
            q(i,m,v)=t1;
        end
    end
end

qlength=t1-xlength-ylength-zlength-ulength;

for i=1:n_ht
    for m=1:n_mt(i)
        for j=1:n_ht
            for n=1:n_mt(j)
                for v=1:le
                    t1=t1+1;
                    l(i,m,j,n,v)=t1;
                end
            end
        end
    end
end

llength=t1-xlength-ylength-zlength-ulength-qlength;

for i=1:tsa
    for m=1:tsb
        t1=t1+1;
        ts(i,m)=t1;
    end
end

tslength=t1-xlength-ylength-zlength-ulength-qlength-llength;

for i=1:tea
    for m=1:teb
        t1=t1+1;
        te(i,m)=t1;
    end
end

telength=t1-xlength-ylength-zlength-ulength-qlength-llength-tslength;

for i=1:ssa
```

```
    for m=1:ssb
        t1=t1+1;
        ss(i,m)=t1;
    end
end

sslength=t1-xlength-ylength-zlength-ulength-qlength-llength-tslength-
telength;

for i=1:sea
    for m=1:seb
        t1=t1+1;
        se(i,m)=t1;
    end
end

selength=t1-xlength-ylength-zlength-ulength-qlength-llength-tslength-
telength-sslength;

%Generate C vector
C=zeros(1, xlength+ylength+zlength+ulength+qlength+llength+telength
+tslength+sslength+selength);

for v=1:n_v
    for i=1:n_ht
        for m=1:n_mt(i)
            for j=1:n_ht
                for n=1:n_mt(j)
                    index1=x(i,m,j,n,v);
                    C(index1)=C(index1)+CS(i,j,v);
                end
            end
        end
    end
end


for i=1:n_ht
    index1=u(i);
    C(index1)=C(index1)-T*EF*R(i)-ET(i);
end

C_const=0;
for i=1:n_ht
    C_const=C_const+T*EF*R(i)+ET(i);
end

%Generate A matrix-------------------------------

A=zeros(100000,xlength+ylength+zlength+ulength+qlength+llength
+telength+tslength+sslength+selength);

%Set Constraints---------------------------------
```

```
%constraint 1
t1=0;
for v=1:n_v
    for i=1:n_ht
        for m=1:n_mt(i)
            t1=t1+1;
            model.sense(t1)='=';
            B(t1)=-W(i,m,v);
            index3=z(i,m,v);
            for j=1:n_ht
                for n=1:n_mt(j)

            index1=x(j,n,i,m,v);
            index2=x(i,m,j,n,v);

            A(t1,index1)=A(t1,index1)+1;
            A(t1,index2)=A(t1,index2)-1;

                end
            end
            A(t1,index3)=A(t1,index3)-1;
        end
    end
end



%constraint 2

t2=t1;
for i=1:n_ht
    for m=1:n_mt(i)
        t2=t2+1;
        model.sense(t2)='=';
        B(t2)=1-sum(W(i,m,:));

        index2=y(i,m);
        A(t2,index2)=A(t2,index2)+1;

        for j=1:n_ht
            for n=1:n_mt(j)
                for v=1:n_v
            index1=x(j,n,i,m,v);
            A(t2,index1)=1;
                end
            end
        end
    end
end

%constraint 3

t3=t2;
for i=1:n_ht
```

```
        for m=2:n_mt(i)
            t3=t3+1;
            model.sense(t3)='>';
            B(t3)=0;
            index1=y(i,m);
            index2=y(i,m-1);

            A(t3,index1)=A(t3,index1)+1;
            A(t3,index2)=A(t3,index2)-1;
        end
    end

    %costraint 4 -changed
    t4=t3;
    for v=1:n_v
        for i=1:n_ht
            for m=1:n_mt(i)
                t4=t4+1;
                model.sense(t4)='=';
                B(t4)=0;
                index3=q(i,m,v);
                A(t4,index3)=A(t4,index3)+J(i)*(1-W(i,m,v));
                for j=1:n_ht
                    for n=1:n_mt(j)
                        index1=l(j,n,i,m,v);
                        index2=l(i,m,j,n,v);
                        A(t4,index1)=A(t4,index1)+1*(1-W(i,m,v));
                        A(t4,index2)=A(t4,index2)-1*(1-W(i,m,v));
                    end
                end
            end
        end
    end

    for v=1:n_v
        for i=1:n_ht
            for m=1:n_mt(i)
                t4=t4+1;
                model.sense(t4)='<';
                B(t4)=CAP(v)*W(i,m,v);
                index3=q(i,m,v);
                A(t4,index3)=A(t4,index3)-J(i)*W(i,m,v);
                for j=1:n_ht
                    for n=1:n_mt(j)
                        index2=l(i,m,j,n,v);
                        A(t4,index2)=A(t4,index2)+1*W(i,m,v);
                    end
                end
            end
        end
    end

    %costraint 5 -changed
    t5=t4;
```

```
for v=1:n_v
    for i=1:n_ht
        for m=1:n_mt(i)
            for j=1:n_ht
                for n=1:n_mt(j)
                    t5=t5+1;
                    model.sense(t5)='<';
                    B(t5)=0;
                    index1=x(i,m,j,n,v);
                    index2=l(i,m,j,n,v);
                    A(t5,index1)=A(t5,index1)-CAP(v);
                    A(t5,index2)=A(t5,index2)+1;
                end
            end
        end
    end
end

%constraint 6 -removed
t6=t5;


%constraint 7
t7=t6;
for v=1:n_v
    for i=1:n_ht
        for m=1:n_mt(i)
            t7=t7+1;
            model.sense(t7)='<';
            B(t7)=W(i,m,v)*QMax(i,m,v);
            index1=q(i,m,v);
            A(t7,index1)=A(t7,index1)+1;
            for j=1:n_ht
                for n=1:n_mt(j)
            index2=x(j,n,i,m,v);
            A(t7,index2)=A(t7,index2)-QMax(i,m,v);
                end
            end
        end
    end
end

%constraint 8
t8=t7;
for i=1:n_ht
    for m=1:n_mt(i)
        t8=t8+1;
        model.sense(t8)='>';
        B(t8)=QMin(i,m);
        index1=y(i,m);
        A(t8,index1)=A(t8,index1)+QMin(i,m);
        for v=1:n_v
        index2=q(i,m,v);
        A(t8,index2)=A(t8,index2)+1;
        end
    end
end

%Time Constraints----------------------------

%constraint 9
t9=t8;
for i=1:n_ht
    for m=1:n_mt(i)
        t9=t9+1;
        model.sense(t9)='=';
        B(t9)=0;
        index1=ts(i,m);
        index2=te(i,m);
        A(t9,index1)=A(t9,index1)+1;
        A(t9,index2)=A(t9,index2)-1;
        for v=1:n_v
        index3=q(i,m,v);
        A(t9,index3)=A(t9,index3)+TQ(i);
        end
        for v=1:n_v
            if hv(v,i)==1
            for j=1:n_ht
                if hv(v,j)==1
                for n=1:n_mt(j)
                index4=x(i,m,j,n,v);
                A(t9,index4)=A(t9,index4)+TSI(i,1);
                end
                end %if
            end
            end %if
        end
    end
end

%constraint 10
t10=t9;
for v=1:n_v
    for i=1:n_ht
        for m=1:n_mt(i)
            for j=1:n_ht
                for n=1:n_mt(j)
                    t10=t10+1;
                    model.sense(t10)='<';
                    B(t10)=M-TS(i,j,v);
                    index1=te(i,m);
                    index2=ts(j,n);
                    index3=x(i,m,j,n,v);
                    A(t10,index1)=A(t10,index1)+1;
                    A(t10,index2)=A(t10,index2)-1;
                    A(t10,index3)=A(t10,index3)+M;
                end
            end
        end
    end
end

%constraint 11
%This constraint was eliminated
t11=t10;

%constraint 12
t12=t11;
for i=1:n_ht
    for m=2:n_mt(i)
        t12=t12+1;
        model.sense(t12)='>';
        B(t12)=TB(i);
        index1=ts(i,m);
        index2=te(i,m-1);
        index3=y(i,m);
        A(t12,index1)=A(t12,index1)+1;
        A(t12,index2)=A(t12,index2)-1;
        A(t12,index3)=A(t12,index3)+TB(i);
    end
end

%constraint 13
t13=t12;
for i=1:n_ht
    for m=1:n_mt(i)
        t13=t13+1;
        model.sense(t13)='>';
        B(t13)=TWS(i,m);
        index1=ts(i,m);
        A(t13,index1)=A(t13,index1)+1;
    end
end

%constraint 14
t14=t13;
for i=1:n_ht
    for m=1:n_mt(i)
        t14=t14+1;
        model.sense(t14)='<';
        B(t14)=TWE(i,m);
        index1=ts(i,m);
        A(t14,index1)=A(t14,index1)+1;
    end
end

%Inventory Constraints-----------------------
%constraint 15
t15=t14;
for i=1:n_ht
    for m=1:n_mt(i)
        t15=t15+1;
```

```
            model.sense(t15)='=';
            B(t15)=0;
            index1=ss(i,m);

            index3=te(i,m);
            index4=ts(i,m);
            index5=se(i,m);
            A(t15,index1)=A(t15,index1)+1;
            for v=1:n_v
                index2=q(i,m,v);
                A(t15,index2)=A(t15,index2)-J(i);
            end
            A(t15,index3)=A(t15,index3)+R(i);
            A(t15,index4)=A(t15,index4)-R(i);
            A(t15,index5)=A(t15,index5)-1;
        end
    end

    %constraint 16
    t16=t15;
    for i=1:n_ht
        for m=2:n_mt(i)
            t16=t16+1;
            model.sense(t16)='=';
            B(t16)=0;
            index1=ss(i,m);
            index3=te(i,m-1);
            index4=ts(i,m);
            index5=se(i,m-1);
            A(t16,index1)=A(t16,index1)-1;
            A(t16,index3)=A(t16,index3)-R(i);
            A(t16,index4)=A(t16,index4)+R(i);
            A(t16,index5)=A(t16,index5)+1;
        end
    end

    %constraint 17
    t17=t16;
    for i=1:n_ht
        for m=1:n_mt(i)
            t17=t17+1;
            model.sense(t17)='>';
            B(t17)=SMin(i);
            index1=ss(i,m);
            A(t17,index1)=A(t17,index1)+1;
        end
    end

    %constraint 18
    t18=t17;
    for i=1:n_ht
        for m=1:n_mt(i)
            t18=t18+1;
            model.sense(t18)='<';
            B(t18)=SMax(i);
            index1=ss(i,m);
            A(t18,index1)=A(t18,index1)+1;
        end
    end

    %constraint 19
    t19=t18;
    for i=1:n_ht
        for m=1:n_mt(i)
            t19=t19+1;
            model.sense(t19)='>';
            B(t19)=SMin(i);
            index1=se(i,m);
            A(t19,index1)=A(t19,index1)+1;
        end
    end

    %constraint 20
    t20=t19;
    for i=1:n_ht
        for m=1:n_mt(i)
            t20=t20+1;
            model.sense(t20)='<';
            B(t20)=SMax(i);
            index1=se(i,m);
            A(t20,index1)=A(t20,index1)+1;
        end
    end

    %constraint 21
    t21=t20;
    for i=1:n_ht
        m=n_mt(i);
            t21=t21+1;
            model.sense(t21)='>';
            B(t21)=0;
            index1=se(i,m);
            index2=te(i,m);
            index3=u(i);
            A(t21,index1)=A(t21,index1)+1;
            A(t21,index2)=A(t21,index2)-R(i);
            A(t21,index3)=A(t21,index3)-SEMin(i)+R(i)*T;
    end

    %constraint 22
    t22=t21;
    for i=1:n_ht
        m=n_mt(i);
            t22=t22+1;
            model.sense(t22)='<';
            B(t22)=SMax(i)-R(i)*T;
            index1=se(i,m);
            index2=te(i,m);
        A(t22,index1)=A(t22,index1)+1;
        A(t22,index2)=A(t22,index2)-R(i);
end

%constraint 23
t23=t22;
for i=1:n_ht
    m=1;
        t23=t23+1;
        model.sense(t23)='=';
        B(t23)=SS(i);
        index1=ss(i,m);
        index2=ts(i,m);
        A(t23,index1)=A(t23,index1)+1;
        A(t23,index2)=A(t23,index2)-R(i);
end




%assignes values to the Gurobi model
model.vtype(1:xlength+ylength+zlength+ulength)='B';%defines binary
 variables
model.vtype(xlength+ylength+zlength+ulength+1:xlength+ylength
+zlength+ulength+qlength+llength+telength+tslength+sslength
+selength)='C';%defines continuous variables

AN=sparse(t23,xlength+ylength+zlength+ulength+qlength+llength+telength
+tslength+sslength+selength);
AN=A(1:t23,1:xlength+ylength+zlength+ulength+qlength+llength+telength
+tslength+sslength+selength);

model.A=sparse(AN);
model.rhs=B;
model.obj=C;
model.objcon=C_const;
model.modelsense='Min';

%Solves the problem in Gurobi
result= gurobi(model)
a=result.x
```

Published with MATLAB® R2018b

A7 - MATLAB Branch-and-Bound algorithm

Script starts on next page.

```
%branch-and-bound

%function branchAndBound(model)

startModel=model;

%Linear Relaxation
for i=1:length(model.vtype);
    model.vtype(i)='C';
end

%Adds 0<=x<=1 constrains for relaxed varaibles
for i=1:length(startModel.vtype)
    if startModel.vtype(i) == 'B'
        [nRow nCol]=size(model.A);
        model.A(nRow+1,:)=0;
        model.A(nRow+1,i)=1;
        model.rhs(nRow+1)=0;
        model.sense(nRow+1)='>';

        [nRow nCol]=size(model.A);
        model.A(nRow+1,:)=0;
        model.A(nRow+1,i)=1;
        model.rhs(nRow+1)=1;
        model.sense(nRow+1)='<';
    end
end

            %Main branch
            clear branch
            branch{1}.parent=[];
            branch{1}.children=[];
            branch{1}.level=1;
            branch{1}.numberLine=[];
            branch{1}.binaryLine=[];
            result = gurobi(model);
            if strcmp(result.status,'INFEASIBLE')
            branch{1}.continuousOptimal=Inf;
            branch{1}.integerOptimal=Inf;
            else
            branch{1}.continuousOptimal=result.objval;
            test=1;
            for i=1:length(startModel.vtype)
                if startModel.vtype(i) == 'B'
                    if result.x(i)==0 || result.x(i)==1
                    else
                        test=0;
                        branch{1}.failVariable=i;
                        branch{1}.integerOptimal=Inf;
                        break
                    end
                end
                end
                if test==1
                    branch{1}.integerOptimal=result.objval;
                end
                end

                if branch{1}.continuousOptimal==Inf
                    branch{1}.endBranch=1;
                else
                    branch{1}.endBranch=0;
                end

bestBrachIntegerOptimal=branch{1}.integerOptimal;
test=0;
level=0;
while level<20
    level=level+1;
    n=length(branch);
    for i=1:n

        if isempty(branch{i}.children) &&
 (branch{i}.continuousOptimal<bestBrachIntegerOptimal) &&
 (branch{i}.endBranch==0)
            %Generate two new branches with old constraints and new
            n=length(branch);
            branch{i}.children=[n+1 n+2];

            %branch 1
            branch{n+1}.parent=i;
            branch{n+1}.children=[];
            branch{n+1}.level=branch{i}.level+1;
            branch{n+1}.numberLine=branch{i}.numberLine;
            branch{n+1}.numberLine(end+1)=branch{i}.failVariable;
            branch{n+1}.binaryLine=branch{i}.binaryLine;
            branch{n+1}.binaryLine(end+1)=0;
            result = branchCalc(model,branch{n+1}.numberLine,branch{n
+1}.binaryLine);
            if strcmp(result.status,'INFEASIBLE')
            branch{n+1}.continuousOptimal=Inf;
            branch{n+1}.integerOptimal=Inf;
            else
            branch{n+1}.continuousOptimal=result.objval;
            test=1;
            for j=1:length(startModel.vtype);
                if startModel.vtype(j) == 'B'
                    if result.x(j)==0 || result.x(j)==1
                    else
                        test=0;
                        branch{n+1}.failVariable=j;
                        branch{n+1}.integerOptimal=Inf;
                        break
                    end
                end
            end
```

```
                if test==1
                    branch{n+1}.integerOptimal=result.objval;
                end
                end

                if (branch{n+1}.continuousOptimal==Inf) || (branch{n
+1}.continuousOptimal>bestBrachIntegerOptimal)
                    branch{n+1}.endBranch=1;
                else
                    branch{n+1}.endBranch=0;
                end

                %branch 2
                branch{n+2}.parent=i;
                branch{n+2}.children=[];
                branch{n+2}.level=branch{i}.level+1;
                branch{n+2}.numberLine=branch{i}.numberLine;
                branch{n+2}.numberLine(end+1)=branch{i}.failVariable;
                branch{n+2}.binaryLine=branch{i}.binaryLine;
                branch{n+2}.binaryLine(end+1)=1;
                result = branchCalc(model,branch{n+2}.numberLine,branch{n
+2}.binaryLine);
                if strcmp(result.status,'INFEASIBLE')
                branch{n+2}.continuousOptimal=Inf;
                branch{n+2}.integerOptimal=Inf;
                else
                branch{n+2}.continuousOptimal=result.objval;
                test=1;
                for j=1:length(startModel.vtype);
                    if startModel.vtype(j) == 'B'
                        if result.x(j)==0 || result.x(j)==1
                        else
                            test=0;
                            branch{n+2}.failVariable=j;
                            branch{n+2}.integerOptimal=Inf;
                            break
                        end
                    end
                end
                if test==1
                    branch{n+2}.integerOptimal=result.objval;
                end
                end

                if (branch{n+2}.continuousOptimal==Inf) || (branch{n
+2}.continuousOptimal>bestBrachIntegerOptimal)
                    branch{n+2}.endBranch=1;
                else
                    branch{n+2}.endBranch=0;
                end

                %update best values
```

```
 bestBrachIntegerOptimal=min(bestBrachIntegerOptimal,branch{n
+1}.integerOptimal);

 bestBrachIntegerOptimal=min(bestBrachIntegerOptimal,branch{n
+2}.integerOptimal);


        end %if no children
    end %for

    %update best lowerbound
            bestLowerBound=Inf;
            for i=1:length(branch)
                if isempty(branch{i}.children)

 bestLowerBound=min(branch{i}.continuousOptimal,bestLowerBound);
                end
            end
            disp('Lower bound: ')
            disp(bestLowerBound)

            disp('Upper bound: ')
            disp(bestBrachIntegerOptimal)

end


function [result] =branchCalc(model,numberLine,binaryLine)
%numberLine is vector with numbers for variables that are fixed
%binaryLine is vector with binary value for variables that are fixed

for i=1:length(numberLine)
    [nRow nCol]=size(model.A);
    variableNumber=numberLine(i);
    binaryValue=binaryLine(i);
    model.A(nRow+1,:)=0;
    model.A(nRow+1,variableNumber)=1;
    model.rhs(nRow+1)=binaryValue;
    model.sense(nRow+1)='=';
end %for

params.outputflag = 0;
result=gurobi(model,params);

end %function
```

Published with MATLAB® R2018b

A8 - Solutions

Script starts on next page.

```
----------------------------------------------------------------------
|                    Optimization results                            |
----------------------------------------------------------------------
|                    Arc-Load Standard                               |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|  12   |     Zones |       Unlimited|  13340 | 13340 |     0.0%      |
----------------------------------------------------------------------
|                      Sailing costs                                 |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|   13340 |   3263 |   7929 |   1118 |   1030 |        0   |       0  |
----------------------------------------------------------------------
|                    Quantity discharged                             |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |  -3000 |  -2700 |  -1500 |  -3000 |    -10200  |    No    |
|    2    |      0 |     10 |      0 |      0 |       10   |    No    |
|    3    |      0 |    900 |      0 |      0 |      900   |    No    |
|    4    |      0 |      0 |      0 |      0 |        0   |    No    |
|    5    |      0 |    500 |      0 |      0 |      500   |    No    |
|    6    |      0 |   1300 |   1400 |      0 |     2700   |    No    |
|    7    |      0 |      0 |   1600 |      0 |     1600   |    No    |
|    8    |      0 |      0 |      0 |   3000 |     3000   |    No    |
|    9    |      0 |      0 |      0 |   1500 |     1500   |    No    |
|   10    |      0 |   1900 |      0 |      0 |     1900   |    No    |
|   11    |      0 |    800 |      0 |      0 |      800   |    No    |
|   12    |   3000 |      0 |      0 |      0 |     3000   |    No    |
Ship total|   3000 |   5410 |   3000 |   4500 |    15910   |          |
----------------------------------------------------------------------
|                      Time of visit                                 |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4| Visit 5|
----------------------------------------------------------------------
|    1    |      0 |     12 |     23 |     95 |    105 |
|    2    |      0 |      0 |      0 |      0 |      0 |
|    3    |      1 |      0 |      0 |      0 |      0 |
|    4    |      0 |      0 |      0 |      0 |      0 |
|    5    |     30 |      0 |      0 |      0 |      0 |
|    6    |     41 |    168 |      0 |      0 |      0 |
|    7    |      0 |     37 |      0 |      0 |      0 |
|    8    |      0 |    115 |      0 |      0 |      0 |
|    9    |    168 |      0 |      0 |      0 |      0 |
|   10    |    124 |      0 |      0 |      0 |      0 |
|   11    |    160 |      0 |      0 |      0 |      0 |
|   12    |    168 |      0 |      0 |      0 |      0 |
-----------------------------------------------------------------------
|                   Optimization results                              |
-----------------------------------------------------------------------
|                   Arc-Flow Standard                                 |
-----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   19  |   Natural |       Unlimited|  11723 | 10683 |     8.9%      |
-----------------------------------------------------------------------
|                     Sailing costs                                   |
-----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
-----------------------------------------------------------------------
|   11723 |   8964 |    801 |   1322 |    636 |        0   |       0  |
-----------------------------------------------------------------------
|                   Quantity discharged                               |
-----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
-----------------------------------------------------------------------
|    1    |  -6049 |      0 |  -1300 |      0 |    -7349   |    No    |
|    2    |    900 |      0 |      0 |      0 |      900   |    No    |
|    3    |    500 |      0 |      0 |      0 |      500   |    No    |
|    4    |      0 |      0 |      0 |    400 |      400   |    No    |
|    5    |    500 |      0 |      0 |      0 |      500   |    No    |
|    6    |   1000 |      0 |      0 |      0 |     1000   |    No    |
|    7    |      0 |      0 |      0 |    800 |      800   |    No    |
|    8    |      0 |      0 |      0 |    300 |      300   |    No    |
|    9    |      0 |      0 |   1300 |      0 |     1300   |    No    |
|   10    |    849 |      0 |      0 |      0 |      849   |    No    |
|   11    |   2300 |      0 |      0 |      0 |     2300   |    No    |
|   12    |      0 |      0 |    600 |      0 |      600   |    No    |
|   13    |    400 |      0 |      0 |      0 |      400   |    No    |
|   14    |   1300 |      0 |      0 |      0 |     1300   |    No    |
|   15    |     -0 |      0 |    900 |      0 |      900   |    No    |
|   16    |    400 |      0 |      0 |      0 |      400   |    No    |
|   17    |    800 |      0 |      0 |      0 |      800   |    No    |
|   18    |      0 |   1500 |      0 |      0 |     1500   |    No    |
|   19    |      0 |    800 |      0 |      0 |      800   |    No    |
Ship total|   8949 |   2300 |   2800 |   1500 |    15549   |          |
-----------------------------------------------------------------------
|                     Time of visit                                   |
-----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
-----------------------------------------------------------------------
|    1    |     66 |     73 |     78 |    116 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |     28 |      0 |      0 |      0 |
|    4    |     24 |      0 |      0 |      0 |
|    5    |     38 |      0 |      0 |      0 |
|    6    |     45 |      0 |      0 |      0 |
|    7    |      4 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |      0 |      0 |      0 |      0 |
|   10    |     71 |      0 |      0 |      0 |
|   11    |     93 |      0 |      0 |      0 |
|   12    |    148 |      0 |      0 |      0 |
|   13    |    134 |      0 |      0 |      0 |
|   14    |    141 |      0 |      0 |      0 |
|   15    |    168 |      0 |      0 |      0 |
|   16    |    162 |      0 |      0 |      0 |
|   17    |    168 |      0 |      0 |      0 |
|   18    |     11 |      0 |      0 |      0 |
|   19    |      0 |      0 |      0 |      0 |
-----------------------------------------------------------------------
|                    Optimization results                             |
-----------------------------------------------------------------------
|                    Arc-Flow Sub-Tour Aggregated                     |
-----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|  19   |   Natural |       Unlimited|  11519 | 10517 |     8.7%      |
-----------------------------------------------------------------------
|                      Sailing costs                                  |
-----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
-----------------------------------------------------------------------
|   11519 |   8530 |    801 |   1551 |    636 |        0   |      -0  |
-----------------------------------------------------------------------
|                    Quantity discharged                              |
-----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
-----------------------------------------------------------------------
|    1    |  -5200 |      0 |  -1700 |      0 |    -6900   |    No    |
|    2    |    900 |      0 |      0 |      0 |      900   |    No    |
|    3    |    500 |      0 |      0 |      0 |      500   |    No    |
|    4    |      0 |      0 |      0 |    400 |      400   |    No    |
|    5    |    500 |      0 |      0 |      0 |      500   |    No    |
|    6    |   1000 |      0 |      0 |      0 |     1000   |    No    |
|    7    |      0 |      0 |      0 |    800 |      800   |    No    |
|    8    |      0 |      0 |      0 |    300 |      300   |    No    |
|    9    |      0 |      0 |   1300 |      0 |     1300   |    No    |
|   10    |      0 |      0 |    400 |      0 |      400   |    No    |
|   11    |   2300 |      0 |      0 |      0 |     2300   |    No    |
|   12    |      0 |      0 |    600 |      0 |      600   |    No    |
|   13    |    400 |      0 |      0 |      0 |      400   |    No    |
|   14    |   1300 |      0 |      0 |      0 |     1300   |    No    |
|   15    |      0 |      0 |    900 |      0 |      900   |    No    |
|   16    |    400 |      0 |      0 |      0 |      400   |    No    |
|   17    |    900 |      0 |      0 |      0 |      900   |    No    |
|   18    |      0 |   1500 |      0 |      0 |     1500   |    No    |
|   19    |      0 |    800 |      0 |      0 |      800   |    No    |
Ship total|   8200 |   2300 |   3200 |   1500 |    15200   |          |
-----------------------------------------------------------------------
|                      Time of visit                                  |
-----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
-----------------------------------------------------------------------
|    1    |     66 |     74 |    106 |    112 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |     28 |      0 |      0 |      0 |
|    4    |    168 |      0 |      0 |      0 |
|    5    |     38 |      0 |      0 |      0 |
|    6    |     45 |      0 |      0 |      0 |
|    7    |      4 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |      0 |      0 |      0 |      0 |
|   10    |     78 |      0 |      0 |      0 |
|   11    |     77 |      0 |      0 |      0 |
|   12    |    148 |      0 |      0 |      0 |
|   13    |    134 |      0 |      0 |      0 |
|   14    |    141 |      0 |      0 |      0 |
|   15    |    168 |      0 |      0 |      0 |
|   16    |    162 |      0 |      0 |      0 |
|   17    |    168 |      0 |      0 |      0 |
|   18    |     11 |      0 |      0 |      0 |
|   19    |      0 |      0 |      0 |      0 |
-----------------------------------------------------------------------
|                    Optimization results                             |
-----------------------------------------------------------------------
|                    Arc-Flow Sub-Tour                                |
-----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|  19  |   Natural |       Unlimited|  11519 |  9878 |     14.2%     |
-----------------------------------------------------------------------
|                      Sailing costs                                  |
-----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
-----------------------------------------------------------------------
|   11519 |   8530 |    801 |   1551 |    636 |        0   |      -0  |
-----------------------------------------------------------------------
|                    Quantity discharged                              |
-----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
-----------------------------------------------------------------------
|    1    |  -5200 |      0 |  -2280 |      0 |    -7480   |    No    |
|    2    |    900 |      0 |      0 |      0 |      900   |    No    |
|    3    |    500 |      0 |      0 |      0 |      500   |    No    |
|    4    |      0 |      0 |      0 |    400 |      400   |    No    |
|    5    |    500 |      0 |      0 |      0 |      500   |    No    |
|    6    |   1100 |      0 |      0 |      0 |     1100   |    No    |
|    7    |      0 |      0 |      0 |    800 |      800   |    No    |
|    8    |      0 |      0 |      0 |    300 |      300   |    No    |
|    9    |      0 |      0 |   1300 |      0 |     1300   |    No    |
|   10    |      0 |      0 |    980 |      0 |      980   |    No    |
|   11    |   2300 |      0 |      0 |      0 |     2300   |    No    |
|   12    |      0 |      0 |    600 |      0 |      600   |    No    |
|   13    |    400 |      0 |      0 |      0 |      400   |    No    |
|   14    |   1300 |      0 |      0 |      0 |     1300   |    No    |
|   15    |      0 |      0 |    900 |      0 |      900   |    No    |
|   16    |    400 |      0 |      0 |      0 |      400   |    No    |
|   17    |    800 |      0 |      0 |      0 |      800   |    No    |
|   18    |      0 |   1529 |      0 |      0 |     1529   |    No    |
|   19    |      0 |   1471 |      0 |      0 |     1471   |    No    |
Ship total|   8200 |   3000 |   3780 |   1500 |    16480   |          |
-----------------------------------------------------------------------
|                      Time of visit                                  |
-----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
-----------------------------------------------------------------------
|    1    |     61 |     66 |    112 |    116 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |     28 |      0 |      0 |      0 |
|    4    |     24 |      0 |      0 |      0 |
|    5    |     38 |      0 |      0 |      0 |
|    6    |     45 |      0 |      0 |      0 |
|    7    |      4 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |      0 |      0 |      0 |      0 |
|   10    |    104 |      0 |      0 |      0 |
|   11    |     81 |      0 |      0 |      0 |
|   12    |    125 |      0 |      0 |      0 |
|   13    |    134 |      0 |      0 |      0 |
|   14    |    141 |      0 |      0 |      0 |
|   15    |    145 |      0 |      0 |      0 |
|   16    |    162 |      0 |      0 |      0 |
|   17    |    168 |      0 |      0 |      0 |
|   18    |     14 |      0 |      0 |      0 |
|   19    |      0 |      0 |      0 |      0 |
-----------------------------------------------------------------------
|                   Optimization results                              |
-----------------------------------------------------------------------
|                   Arc-Flow Time Windows                             |
-----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   19  |   Natural |       Unlimited|  12213 | 10536 |     13.7%     |
-----------------------------------------------------------------------
|                     Sailing costs                                   |
-----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
-----------------------------------------------------------------------
|   12213 |   7562 |   1619 |   2754 |    278 |        0   |       0  |
-----------------------------------------------------------------------
|                   Quantity discharged                               |
-----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
-----------------------------------------------------------------------
|    1    |  -4800 |      0 |  -1800 |     -0 |    -6600   |    No    |
|    2    |    900 |      0 |      0 |      0 |      900   |    No    |
|    3    |    500 |      0 |      0 |      0 |      500   |    No    |
|    4    |    400 |      0 |      0 |      0 |      400   |    No    |
|    5    |    500 |      0 |      0 |      0 |      500   |    No    |
|    6    |      0 |      0 |      0 |   1200 |     1200   |    No    |
|    7    |      0 |      0 |    800 |      0 |      800   |    No    |
|    8    |      0 |      0 |      0 |    300 |      300   |    No    |
|    9    |      0 |      0 |   1300 |      0 |     1300   |    No    |
|   10    |    400 |      0 |      0 |      0 |      400   |    No    |
|   11    |   2300 |      0 |      0 |      0 |     2300   |    No    |
|   12    |    600 |      0 |      0 |      0 |      600   |    No    |
|   13    |      0 |      0 |    400 |      0 |      400   |    No    |
|   14    |   1300 |      0 |      0 |      0 |     1300   |    No    |
|   15    |    900 |      0 |      0 |      0 |      900   |    No    |
|   16    |      0 |    400 |      0 |      0 |      400   |    No    |
|   17    |      0 |      0 |    800 |      0 |      800   |    No    |
|   18    |      0 |   1500 |      0 |      0 |     1500   |    No    |
|   19    |      0 |    800 |      0 |      0 |      800   |    No    |
Ship total|   7800 |   2700 |   3300 |   1500 |    15300   |          |
-----------------------------------------------------------------------
|                     Time of visit                                   |
-----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
-----------------------------------------------------------------------
|    1    |     19 |     58 |     62 |     96 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |     28 |      0 |      0 |      0 |
|    4    |     35 |      0 |      0 |      0 |
|    5    |     41 |      0 |      0 |      0 |
|    6    |      8 |      0 |      0 |      0 |
|    7    |     33 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |      0 |      0 |      0 |      0 |
|   10    |     57 |      0 |      0 |      0 |
|   11    |     74 |      0 |      0 |      0 |
|   12    |    115 |      0 |      0 |      0 |
|   13    |     75 |      0 |      0 |      0 |
|   14    |    127 |      0 |      0 |      0 |
|   15    |    144 |      0 |      0 |      0 |
|   16    |     29 |      0 |      0 |      0 |
|   17    |     98 |      0 |      0 |      0 |
|   18    |     11 |      0 |      0 |      0 |
|   19    |      0 |      0 |      0 |      0 |
----------------------------------------------------------------------
|                    Optimization results                            |
----------------------------------------------------------------------
|                    Arc-Flow Clique Inequality                      |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   19  |   Natural |         Limited|  11859 | 10340 |     12.8%     |
----------------------------------------------------------------------
|                      Sailing costs                                 |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|   11859 |   2888 |    801 |    287 |      0 |        0   |    7883  |
----------------------------------------------------------------------
|                    Quantity discharged                             |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |  -5300 |      0 |  -1300 |  -1500 |    -8100   |    No    |
|    2    |    900 |      0 |      0 |      0 |      900   |    No    |
|    3    |    500 |      0 |      0 |      0 |      500   |    No    |
|    4    |    400 |      0 |      0 |      0 |      400   |    No    |
|    5    |      0 |      0 |      0 |      0 |        0   |    No    |
|    6    |      0 |      0 |      0 |      0 |        0   |    No    |
|    7    |      0 |      0 |      0 |      0 |        0   |    No    |
|    8    |      0 |      0 |      0 |    300 |      300   |    No    |
|    9    |      0 |      0 |   1300 |      0 |     1300   |    No    |
|   10    |      0 |      0 |      0 |      0 |        0   |    No    |
|   11    |      0 |      0 |      0 |      0 |        0   |    No    |
|   12    |      0 |      0 |      0 |      0 |        0   |    No    |
|   13    |      0 |      0 |      0 |      0 |        0   |    No    |
|   14    |      0 |      0 |      0 |      0 |        0   |    No    |
|   15    |      0 |      0 |      0 |      0 |        0   |    No    |
|   16    |      0 |      0 |      0 |      0 |        0   |    No    |
|   17    |      0 |      0 |      0 |      0 |        0   |    No    |
|   18    |      0 |   1500 |      0 |      0 |     1500   |    No    |
|   19    |      0 |    800 |      0 |      0 |      800   |    No    |
Ship total|   1800 |   2300 |   1300 |    300 |     5700   |          |
----------------------------------------------------------------------
|                      Time of visit                                 |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
----------------------------------------------------------------------
|    1    |     94 |    119 |    131 |    136 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |     28 |      0 |      0 |      0 |
|    4    |     71 |      0 |      0 |      0 |
|    5    |    168 |      0 |      0 |      0 |
|    6    |    159 |      0 |      0 |      0 |
|    7    |     82 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |      0 |      0 |      0 |      0 |
|   10    |    132 |      0 |      0 |      0 |
|   11    |    107 |      0 |      0 |      0 |
|   12    |    152 |      0 |      0 |      0 |
|   13    |    143 |      0 |      0 |      0 |
|   14    |    148 |      0 |      0 |      0 |
|   15    |    168 |      0 |      0 |      0 |
|   16    |    163 |      0 |      0 |      0 |
|   17    |    168 |      0 |      0 |      0 |
|   18    |     11 |      0 |      0 |      0 |
|   19    |      0 |      0 |      0 |      0 |
----------------------------------------------------------------------
|                    Optimization results                            |
----------------------------------------------------------------------
|                    Arc-Load Standard Formulation                   |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|  19  |   Natural |       Unlimited|  16900 |  6771 |     59.9%     |
----------------------------------------------------------------------
|                      Sailing costs                                 |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|   16900 |      0 |   8238 |   1967 |   1295 |      900   |    4500  |
----------------------------------------------------------------------
|                    Quantity discharged                             |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |      0 |  -6000 |  -1200 |  -1500 |    -8700   |    No    |
|    2    |   1655 |      0 |      0 |      0 |     1655   |    No    |
|    3    |      0 |      0 |    500 |      0 |      500   |    No    |
|    4    |      0 |      0 |    400 |      0 |      400   |    No    |
|    5    |      0 |      0 |    500 |      0 |      500   |    No    |
|    6    |      0 |      0 |      0 |   1500 |     1500   |    No    |
|    7    |      0 |      0 |      0 |    800 |      800   |    No    |
|    8    |      0 |      0 |      0 |    300 |      300   |    No    |
|    9    |      0 |      0 |   1300 |      0 |     1300   |    No    |
|   10    |      0 |      0 |      0 |    400 |      400   |    No    |
|   11    |      0 |   3000 |      0 |      0 |     3000   |    No    |
|   12    |      0 |    600 |      0 |      0 |      600   |    No    |
|   13    |      0 |    400 |      0 |      0 |      400   |    No    |
|   14    |      0 |   1300 |      0 |      0 |     1300   |    No    |
|   15    |      0 |     10 |      0 |      0 |       10   |    Yes   |
|   16    |      0 |    400 |      0 |      0 |      400   |    No    |
|   17    |      0 |    890 |      0 |      0 |      890   |    No    |
|   18    |      0 |   1500 |      0 |      0 |     1500   |    No    |
|   19    |      0 |    800 |      0 |      0 |      800   |    No    |
Ship total|   1655 |   8900 |   2700 |   3000 |    16255   |          |
----------------------------------------------------------------------
|                      Time of visit                                 |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
----------------------------------------------------------------------
|    1    |     57 |     97 |    114 |    120 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |    154 |      0 |      0 |      0 |
|    4    |    161 |      0 |      0 |      0 |
|    5    |    168 |      0 |      0 |      0 |
|    6    |    136 |      0 |      0 |      0 |
|    7    |      4 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |      0 |      0 |      0 |      0 |
|   10    |    108 |      0 |      0 |      0 |
|   11    |     72 |      0 |      0 |      0 |
|   12    |     43 |      0 |      0 |      0 |
|   13    |    119 |      0 |      0 |      0 |
|   14    |    125 |      0 |      0 |      0 |
|   15    |    168 |      0 |      0 |      0 |
|   16    |    147 |      0 |      0 |      0 |
|   17    |    153 |      0 |      0 |      0 |
|   18    |     11 |      0 |      0 |      0 |
|   19    |      0 |      0 |      0 |      0 |
----------------------------------------------------------------------
|                   Optimization results                             |
----------------------------------------------------------------------
|                   Arc-Load Sub-Tour Aggreagated                    |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|  19  |   Natural |       Unlimited|  12125 |  6802 |     43.9%      |
----------------------------------------------------------------------
|                     Sailing costs                                  |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|   12125 |   8371 |   1619 |    840 |   1295 |        0   |       0  |
----------------------------------------------------------------------
|                   Quantity discharged                              |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |  -4600 |     -0 |  -1500 |  -1500 |    -7600   |    No    |
|    2    |    900 |      0 |      0 |      0 |      900   |    No    |
|    3    |    500 |      0 |      0 |      0 |      500   |    No    |
|    4    |    400 |      0 |      0 |      0 |      400   |    No    |
|    5    |    500 |      0 |      0 |      0 |      500   |    No    |
|    6    |      0 |      0 |      0 |   1100 |     1100   |    No    |
|    7    |      0 |      0 |      0 |    800 |      800   |    No    |
|    8    |      0 |      0 |      0 |    300 |      300   |    No    |
|    9    |      0 |      0 |   1300 |      0 |     1300   |    No    |
|   10    |      0 |      0 |      0 |    400 |      400   |    No    |
|   11    |   2300 |      0 |      0 |      0 |     2300   |    No    |
|   12    |      0 |      0 |    600 |      0 |      600   |    No    |
|   13    |      0 |      0 |    900 |      0 |      900   |    No    |
|   14    |   1300 |      0 |      0 |      0 |     1300   |    No    |
|   15    |    900 |      0 |      0 |      0 |      900   |    No    |
|   16    |      0 |    700 |      0 |      0 |      700   |    No    |
|   17    |    800 |      0 |      0 |      0 |      800   |    No    |
|   18    |      0 |   1500 |      0 |      0 |     1500   |    No    |
|   19    |      0 |    800 |      0 |      0 |      800   |    No    |
Ship total|   7600 |   3000 |   2800 |   2600 |    16000   |          |
----------------------------------------------------------------------
|                     Time of visit                                  |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
----------------------------------------------------------------------
|    1    |     94 |    100 |    116 |    122 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |     72 |      0 |      0 |      0 |
|    4    |     77 |      0 |      0 |      0 |
|    5    |     83 |      0 |      0 |      0 |
|    6    |    168 |      0 |      0 |      0 |
|    7    |     79 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |      0 |      0 |      0 |      0 |
|   10    |    102 |      0 |      0 |      0 |
|   11    |    109 |      0 |      0 |      0 |
|   12    |    163 |      0 |      0 |      0 |
|   13    |    168 |      0 |      0 |      0 |
|   14    |    147 |      0 |      0 |      0 |
|   15    |    157 |      0 |      0 |      0 |
|   16    |    168 |      0 |      0 |      0 |
|   17    |    168 |      0 |      0 |      0 |
|   18    |    154 |      0 |      0 |      0 |
|   19    |      0 |      0 |      0 |      0 |
----------------------------------------------------------------------
|                    Optimization results                            |
----------------------------------------------------------------------
|                    Arc-Load Time Window                            |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|  19   |   Natural |       Unlimited|  15258 |  7039 |     53.9%     |
----------------------------------------------------------------------
|                      Sailing costs                                 |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|   15258 |   9331 |   2913 |   1388 |   1627 |        0   |       0  |
----------------------------------------------------------------------
|                    Quantity discharged                             |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |  -3800 |      0 |  -1300 |  -1500 |    -6600   |    No    |
|    2    |    900 |      0 |      0 |      0 |      900   |    No    |
|    3    |    500 |      0 |      0 |      0 |      500   |    No    |
|    4    |    400 |      0 |      0 |      0 |      400   |    No    |
|    5    |      0 |      0 |    500 |      0 |      500   |    No    |
|    6    |      0 |      0 |   1000 |      0 |     1000   |    No    |
|    7    |      0 |      0 |      0 |    800 |      800   |    No    |
|    8    |      0 |      0 |      0 |    300 |      300   |    No    |
|    9    |      0 |      0 |   1300 |      0 |     1300   |    No    |
|   10    |      0 |      0 |      0 |    400 |      400   |    No    |
|   11    |   2300 |      0 |      0 |      0 |     2300   |    No    |
|   12    |      0 |    600 |      0 |      0 |      600   |    No    |
|   13    |      0 |      0 |      0 |    400 |      400   |    No    |
|   14    |      0 |   1300 |      0 |      0 |     1300   |    No    |
|   15    |      0 |      0 |      0 |   1100 |     1100   |    No    |
|   16    |    400 |      0 |      0 |      0 |      400   |    No    |
|   17    |    800 |      0 |      0 |      0 |      800   |    No    |
|   18    |   1500 |      0 |      0 |      0 |     1500   |    No    |
|   19    |      0 |    800 |      0 |      0 |      800   |    No    |
Ship total|   6800 |   2700 |   2800 |   3000 |    15300   |          |
----------------------------------------------------------------------
|                      Time of visit                                 |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
----------------------------------------------------------------------
|    1    |     19 |     56 |     83 |     89 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |     28 |      0 |      0 |      0 |
|    4    |     35 |      0 |      0 |      0 |
|    5    |     44 |      0 |      0 |      0 |
|    6    |     53 |      0 |      0 |      0 |
|    7    |     60 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |      0 |      0 |      0 |      0 |
|   10    |     77 |      0 |      0 |      0 |
|   11    |     66 |      0 |      0 |      0 |
|   12    |    144 |      0 |      0 |      0 |
|   13    |    130 |      0 |      0 |      0 |
|   14    |    125 |      0 |      0 |      0 |
|   15    |    144 |      0 |      0 |      0 |
|   16    |    118 |      0 |      0 |      0 |
|   17    |    144 |      0 |      0 |      0 |
|   18    |    128 |      0 |      0 |      0 |
|   19    |      0 |      0 |      0 |      0 |
----------------------------------------------------------------------
|                    Optimization results                            |
----------------------------------------------------------------------
|                    Arc-Load Clique Inequialities                   |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   19   |   Natural |       Unlimited|  12133 |  8161 |     32.7%     |
----------------------------------------------------------------------
|                      Sailing costs                                 |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|   12133 |      0 |    801 |    840 |    871 |        0   |    9621  |
----------------------------------------------------------------------
|                    Quantity discharged                             |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |  -5300 |      0 |  -1200 |   -600 |    -7100   |    No    |
|    2    |    900 |      0 |      0 |      0 |      900   |    No    |
|    3    |      0 |      0 |      0 |      0 |        0   |    No    |
|    4    |      0 |      0 |      0 |      0 |        0   |    No    |
|    5    |      0 |      0 |      0 |    500 |      500   |    No    |
|    6    |      0 |      0 |      0 |      0 |        0   |    No    |
|    7    |      0 |      0 |      0 |      0 |        0   |    No    |
|    8    |      0 |      0 |      0 |    300 |      300   |    No    |
|    9    |      0 |      0 |   1300 |      0 |     1300   |    No    |
|   10    |      0 |      0 |      0 |      0 |        0   |    No    |
|   11    |      0 |      0 |      0 |      0 |        0   |    No    |
|   12    |      0 |      0 |    600 |      0 |      600   |    No    |
|   13    |      0 |      0 |    400 |      0 |      400   |    No    |
|   14    |      0 |      0 |      0 |      0 |        0   |    No    |
|   15    |      0 |      0 |      0 |      0 |        0   |    No    |
|   16    |      0 |      0 |      0 |      0 |        0   |    No    |
|   17    |      0 |      0 |      0 |      0 |        0   |    No    |
|   18    |      0 |   1500 |      0 |      0 |     1500   |    No    |
|   19    |      0 |    800 |      0 |      0 |      800   |    No    |
Ship total|    900 |   2300 |   2300 |    800 |     6300   |          |
----------------------------------------------------------------------
|                      Time of visit                                 |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
----------------------------------------------------------------------
|    1    |     15 |     65 |    112 |    121 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |    168 |      0 |      0 |      0 |
|    4    |    161 |      0 |      0 |      0 |
|    5    |    153 |      0 |      0 |      0 |
|    6    |     36 |      0 |      0 |      0 |
|    7    |     48 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |      0 |      0 |      0 |      0 |
|   10    |      9 |      0 |      0 |      0 |
|   11    |     80 |      0 |      0 |      0 |
|   12    |    136 |      0 |      0 |      0 |
|   13    |    150 |      0 |      0 |      0 |
|   14    |    135 |      0 |      0 |      0 |
|   15    |    152 |      0 |      0 |      0 |
|   16    |    168 |      0 |      0 |      0 |
|   17    |    168 |      0 |      0 |      0 |
|   18    |     11 |      0 |      0 |      0 |
|   19    |      0 |      0 |      0 |      0 |
----------------------------------------------------------------------
|                    Optimization results                            |
----------------------------------------------------------------------
|                    Arc-Load Standard                               |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|  24  |    KMeans |       Unlimited|  23596 |  7103 |     69.9%     |
----------------------------------------------------------------------
|                      Sailing costs                                 |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|   23596 |   4979 |    145 |   2208 |   4863 |     1900   |    9500  |
----------------------------------------------------------------------
|                    Quantity discharged                             |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |  -3010 |     -0 |  -1500 |  -2900 |    -7410   |    No    |
|    2    |      0 |    400 |      0 |      0 |      400   |    No    |
|    3    |      0 |   1100 |      0 |      0 |     1100   |    No    |
|    4    |    500 |      0 |      0 |      0 |      500   |    No    |
|    5    |    200 |      0 |      0 |      0 |      200   |    Yes   |
|    6    |    500 |      0 |      0 |      0 |      500   |    No    |
|    7    |   1000 |      0 |      0 |      0 |     1000   |    No    |
|    8    |      0 |      0 |    800 |      0 |      800   |    No    |
|    9    |    800 |      0 |      0 |      0 |      800   |    No    |
|   10    |      0 |      0 |    600 |      0 |      600   |    No    |
|   11    |      0 |      0 |      0 |    500 |      500   |    No    |
|   12    |      0 |      0 |    100 |      0 |      100   |    No    |
|   13    |   1800 |      0 |      0 |      0 |     1800   |    No    |
|   14    |      0 |      0 |      0 |    500 |      500   |    No    |
|   15    |      0 |      0 |      0 |    425 |      425   |    No    |
|   16    |      0 |      0 |      0 |    436 |      436   |    No    |
|   17    |   1041 |     -0 |      0 |      0 |     1041   |    No    |
|   18    |      0 |      0 |      0 |    900 |      900   |    No    |
|   19    |      0 |      0 |      0 |    300 |      300   |    No    |
|   20    |      0 |      0 |    700 |      0 |      700   |    No    |
|   21    |      0 |      0 |      0 |    400 |      400   |    No    |
|   22    |      0 |      0 |    800 |      0 |      800   |    No    |
|   23    |      0 |      0 |      0 |      0 |        0   |    Yes   |
|   24    |      0 |      0 |      0 |    800 |      800   |    No    |
Ship total|   5841 |   1500 |   3000 |   4261 |    14602   |          |
----------------------------------------------------------------------
|                      Time of visit                                 |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
----------------------------------------------------------------------
|    1    |      0 |     53 |     59 |    113 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |    164 |      0 |      0 |      0 |
|    4    |    112 |      0 |      0 |      0 |
|    5    |    107 |      0 |      0 |      0 |
|    6    |    101 |      0 |      0 |      0 |
|    7    |     89 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |     78 |      0 |      0 |      0 |
|   10    |     11 |      0 |      0 |      0 |
|   11    |     63 |      0 |      0 |      0 |
|   12    |     21 |      0 |      0 |      0 |
|   13    |     19 |      0 |      0 |      0 |
|   14    |      0 |      0 |      0 |      0 |
|   15    |     34 |      0 |      0 |      0 |
|   16    |     41 |      0 |      0 |      0 |
|   17    |     41 |      0 |      0 |      0 |
|   18    |     88 |      0 |      0 |      0 |
|   19    |    142 |      0 |      0 |      0 |
|   20    |    150 |      0 |      0 |      0 |
|   21    |    150 |      0 |      0 |      0 |
|   22    |    168 |      0 |      0 |      0 |
|   23    |      0 |      0 |      0 |      0 |
|   24    |    168 |      0 |      0 |      0 |
----------------------------------------------------------------------
|                    Optimization results                            |
----------------------------------------------------------------------
|                    Arc-Load Clique Inequalities                    |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|  24  |    KMeans |       Unlimited|  23579 |  6175 |     73.8%     |
----------------------------------------------------------------------
|                      Sailing costs                                 |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|   23579 |  10827 |    145 |   2345 |   2462 |     1300   |    6500  |
----------------------------------------------------------------------
|                    Quantity discharged                             |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |  -4810 |     -0 |  -1500 |  -1500 |    -7810   |    No    |
|    2    |      0 |    400 |      0 |      0 |      400   |    No    |
|    3    |      0 |    500 |      0 |      0 |      500   |    No    |
|    4    |    500 |      0 |      0 |      0 |      500   |    No    |
|    5    |     10 |      0 |      0 |      0 |       10   |    No    |
|    6    |    500 |      0 |      0 |      0 |      500   |    No    |
|    7    |   1190 |      0 |      0 |      0 |     1190   |    No    |
|    8    |      0 |      0 |    800 |      0 |      800   |    No    |
|    9    |    800 |      0 |      0 |      0 |      800   |    No    |
|   10    |      0 |     -0 |    600 |      0 |      600   |    No    |
|   11    |      0 |     -0 |      0 |    500 |      500   |    No    |
|   12    |      0 |      0 |      0 |    100 |      100   |    No    |
|   13    |   1800 |      0 |      0 |      0 |     1800   |    No    |
|   14    |      0 |      0 |      0 |    500 |      500   |    No    |
|   15    |      0 |      0 |      0 |    300 |      300   |    No    |
|   16    |      0 |      0 |    300 |      0 |      300   |    No    |
|   17    |    700 |      0 |      0 |      0 |      700   |    No    |
|   18    |    100 |      0 |      0 |      0 |      100   |    Yes   |
|   19    |      0 |      0 |      0 |    300 |      300   |    No    |
|   20    |    700 |      0 |      0 |      0 |      700   |    No    |
|   21    |      0 |      0 |    400 |      0 |      400   |    No    |
|   22    |      0 |      0 |    800 |      0 |      800   |    No    |
|   23    |   1500 |      0 |      0 |      0 |     1500   |    No    |
|   24    |      0 |      0 |      0 |    900 |      900   |    No    |
Ship total|   7800 |    900 |   2900 |   2600 |    14200   |          |
----------------------------------------------------------------------
|                      Time of visit                                 |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
----------------------------------------------------------------------
|    1    |      0 |     16 |     60 |     66 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |      3 |      0 |      0 |      0 |
|    4    |     32 |      0 |      0 |      0 |
|    5    |     27 |      0 |      0 |      0 |
|    6    |     41 |      0 |      0 |      0 |
|    7    |     12 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |     53 |      0 |      0 |      0 |
|   10    |     49 |      0 |      0 |      0 |
|   11    |      9 |      0 |      0 |      0 |
|   12    |      5 |      0 |      0 |      0 |
|   13    |     81 |      0 |      0 |      0 |
|   14    |      0 |      0 |      0 |      0 |
|   15    |     30 |      0 |      0 |      0 |
|   16    |     76 |      0 |      0 |      0 |
|   17    |    143 |      0 |      0 |      0 |
|   18    |    134 |      0 |      0 |      0 |
|   19    |     48 |      0 |      0 |      0 |
|   20    |    124 |      0 |      0 |      0 |
|   21    |    168 |      0 |      0 |      0 |
|   22    |    101 |      0 |      0 |      0 |
|   23    |    168 |      0 |      0 |      0 |
|   24    |     72 |      0 |      0 |      0 |
-----------------------------------------------------------------------
|                    Optimization results                             |
-----------------------------------------------------------------------
|                    Arc-Load Sub-Tour                                |
-----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   24  |    KMeans |       Unlimited|  19498 |  6833 |     65.0%     |
-----------------------------------------------------------------------
|                      Sailing costs                                  |
-----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
-----------------------------------------------------------------------
|   19498 |   3033 |   3868 |   2105 |   2892 |     1900   |    5700  |
-----------------------------------------------------------------------
|                    Quantity discharged                              |
-----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
-----------------------------------------------------------------------
|    1    |  -1410 |      0 |  -1400 |  -1600 |    -4410   |    No    |
|    2    |      0 |    400 |      0 |      0 |      400   |    No    |
|    3    |      0 |    500 |      0 |      0 |      500   |    No    |
|    4    |      0 |    500 |      0 |      0 |      500   |    No    |
|    5    |      0 |     10 |      0 |      0 |       10   |    Yes   |
|    6    |      0 |    590 |      0 |      0 |      590   |    No    |
|    7    |      0 |   1000 |      0 |      0 |     1000   |    No    |
|    8    |      0 |      0 |    800 |      0 |      800   |    No    |
|    9    |      0 |      0 |      0 |    800 |      800   |    No    |
|   10    |      0 |      0 |    600 |      0 |      600   |    No    |
|   11    |      0 |      0 |      0 |    500 |      500   |    No    |
|   12    |      0 |      0 |      0 |    100 |      100   |    No    |
|   13    |   1800 |      0 |      0 |      0 |     1800   |    No    |
|   14    |      0 |      0 |      0 |    500 |      500   |    No    |
|   15    |    300 |      0 |      0 |      0 |      300   |    No    |
|   16    |      0 |      0 |    300 |      0 |      300   |    No    |
|   17    |    700 |      0 |      0 |      0 |      700   |    No    |
|   18    |    900 |      0 |      0 |      0 |      900   |    No    |
|   19    |      0 |      0 |    300 |      0 |      300   |    No    |
|   20    |    700 |      0 |      0 |      0 |      700   |    No    |
|   21    |      0 |      0 |      0 |    400 |      400   |    No    |
|   22    |      0 |      0 |    800 |      0 |      800   |    No    |
|   23    |      0 |      0 |      0 |      0 |        0   |    Yes   |
|   24    |      0 |      0 |      0 |    800 |      800   |    No    |
Ship total|   4400 |   3000 |   2800 |   3100 |    13300   |          |
-----------------------------------------------------------------------
|                      Time of visit                                  |
-----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
-----------------------------------------------------------------------
|    1    |      0 |     19 |     25 |     26 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |      3 |      0 |      0 |      0 |
|    4    |     28 |      0 |      0 |      0 |
|    5    |     34 |      0 |      0 |      0 |
|    6    |     38 |      0 |      0 |      0 |
|    7    |     44 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |     37 |      0 |      0 |      0 |
|   10    |     10 |      0 |      0 |      0 |
|   11    |     30 |      0 |      0 |      0 |
|   12    |     49 |      0 |      0 |      0 |
|   13    |     16 |      0 |      0 |      0 |
|   14    |      0 |      0 |      0 |      0 |
|   15    |     38 |      0 |      0 |      0 |
|   16    |     35 |      0 |      0 |      0 |
|   17    |     45 |      0 |      0 |      0 |
|   18    |     52 |      0 |      0 |      0 |
|   19    |     50 |      0 |      0 |      0 |
|   20    |     61 |      0 |      0 |      0 |
|   21    |     84 |      0 |      0 |      0 |
|   22    |     61 |      0 |      0 |      0 |
|   23    |      0 |      0 |      0 |      0 |
|   24    |    102 |      0 |      0 |      0 |
-----------------------------------------------------------------------
|                    Optimization results                             |
-----------------------------------------------------------------------
|                    Arc-Load Sub-Tour Aggregated                     |
-----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   24  |    KMeans |       Unlimited|  19498 |  6762 |    65.3%      |
-----------------------------------------------------------------------
|                      Sailing costs                                  |
-----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
-----------------------------------------------------------------------
|   19498 |   3033 |   3868 |   2105 |   2892 |     1900   |    5700  |
-----------------------------------------------------------------------
|                    Quantity discharged                              |
-----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
-----------------------------------------------------------------------
|    1    |  -1410 |      0 |  -1400 |  -1600 |    -4410   |    No    |
|    2    |      0 |    400 |      0 |      0 |      400   |    No    |
|    3    |      0 |    500 |      0 |      0 |      500   |    No    |
|    4    |      0 |    500 |      0 |      0 |      500   |    No    |
|    5    |      0 |     10 |      0 |      0 |       10   |    Yes   |
|    6    |      0 |    590 |      0 |      0 |      590   |    No    |
|    7    |      0 |   1000 |      0 |      0 |     1000   |    No    |
|    8    |      0 |      0 |    800 |      0 |      800   |    No    |
|    9    |      0 |      0 |      0 |    800 |      800   |    No    |
|   10    |      0 |      0 |    600 |      0 |      600   |    No    |
|   11    |      0 |      0 |      0 |    500 |      500   |    No    |
|   12    |      0 |      0 |      0 |    100 |      100   |    No    |
|   13    |   1800 |      0 |      0 |      0 |     1800   |    No    |
|   14    |      0 |      0 |      0 |    500 |      500   |    No    |
|   15    |    300 |      0 |      0 |      0 |      300   |    No    |
|   16    |      0 |      0 |    300 |      0 |      300   |    No    |
|   17    |    700 |      0 |      0 |      0 |      700   |    No    |
|   18    |    900 |      0 |      0 |      0 |      900   |    No    |
|   19    |      0 |      0 |    300 |      0 |      300   |    No    |
|   20    |    700 |      0 |      0 |      0 |      700   |    No    |
|   21    |      0 |      0 |      0 |    400 |      400   |    No    |
|   22    |      0 |      0 |    800 |      0 |      800   |    No    |
|   23    |      0 |      0 |      0 |      0 |        0   |    Yes   |
|   24    |      0 |      0 |      0 |    800 |      800   |    No    |
Ship total|   4400 |   3000 |   2800 |   3100 |    13300   |          |
-----------------------------------------------------------------------
|                      Time of visit                                  |
-----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
-----------------------------------------------------------------------
|    1    |      0 |     19 |     25 |     26 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |      3 |      0 |      0 |      0 |
|    4    |     28 |      0 |      0 |      0 |
|    5    |     34 |      0 |      0 |      0 |
|    6    |     38 |      0 |      0 |      0 |
|    7    |     44 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |     37 |      0 |      0 |      0 |
|   10    |     10 |      0 |      0 |      0 |
|   11    |     30 |      0 |      0 |      0 |
|   12    |     49 |      0 |      0 |      0 |
|   13    |     16 |      0 |      0 |      0 |
|   14    |      0 |      0 |      0 |      0 |
|   15    |     38 |      0 |      0 |      0 |
|   16    |     35 |      0 |      0 |      0 |
|   17    |     45 |      0 |      0 |      0 |
|   18    |     52 |      0 |      0 |      0 |
|   19    |     50 |      0 |      0 |      0 |
|   20    |     61 |      0 |      0 |      0 |
|   21    |     84 |      0 |      0 |      0 |
|   22    |     61 |      0 |      0 |      0 |
|   23    |      0 |      0 |      0 |      0 |
|   24    |    102 |      0 |      0 |      0 |
----------------------------------------------------------------------
|                   Optimization results                              |
----------------------------------------------------------------------
|                   Arc-Flow Sub-Tour Elimination                     |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|  24  |    KMeans |       Unlimited|  14727 | 12044 |     18.2%      |
----------------------------------------------------------------------
|                     Sailing costs                                   |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|   14727 |   9333 |    145 |   2638 |   2610 |        0   |       0  |
----------------------------------------------------------------------
|                   Quantity discharged                               |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |  -5310 |      0 |  -1500 |  -1500 |    -8310   |    No    |
|    2    |      0 |    400 |      0 |      0 |      400   |    No    |
|    3    |      0 |    500 |      0 |      0 |      500   |    No    |
|    4    |      0 |      0 |      0 |    500 |      500   |    No    |
|    5    |      0 |      0 |      0 |    400 |      400   |    No    |
|    6    |      0 |      0 |      0 |    500 |      500   |    No    |
|    7    |   1000 |      0 |      0 |      0 |     1000   |    No    |
|    8    |      0 |      0 |    800 |      0 |      800   |    No    |
|    9    |    800 |      0 |      0 |      0 |      800   |    No    |
|   10    |      0 |      0 |    700 |      0 |      700   |    No    |
|   11    |    500 |      0 |      0 |      0 |      500   |    No    |
|   12    |      0 |      0 |      0 |    100 |      100   |    No    |
|   13    |   1800 |      0 |      0 |      0 |     1800   |    No    |
|   14    |      0 |      0 |      0 |    500 |      500   |    No    |
|   15    |      0 |      0 |      0 |    300 |      300   |    No    |
|   16    |    300 |      0 |      0 |      0 |      300   |    No    |
|   17    |      0 |      0 |      0 |    700 |      700   |    No    |
|   18    |    900 |      0 |      0 |      0 |      900   |    No    |
|   19    |    300 |      0 |      0 |      0 |      300   |    No    |
|   20    |      0 |      0 |    700 |      0 |      700   |    No    |
|   21    |    400 |      0 |      0 |      0 |      400   |    No    |
|   22    |    800 |      0 |      0 |      0 |      800   |    No    |
|   23    |   1500 |      0 |      0 |      0 |     1500   |    No    |
|   24    |      0 |      0 |    800 |      0 |      800   |    No    |
Ship total|   8300 |    900 |   3000 |   3000 |    15200   |          |
----------------------------------------------------------------------
|                     Time of visit                                   |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
----------------------------------------------------------------------
|    1    |      0 |     33 |     39 |     51 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |      3 |      0 |      0 |      0 |
|    4    |    168 |      0 |      0 |      0 |
|    5    |    161 |      0 |      0 |      0 |
|    6    |     66 |      0 |      0 |      0 |
|    7    |     78 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |     67 |      0 |      0 |      0 |
|   10    |     11 |      0 |      0 |      0 |
|   11    |     95 |      0 |      0 |      0 |
|   12    |     46 |      0 |      0 |      0 |
|   13    |     34 |      0 |      0 |      0 |
|   14    |      0 |      0 |      0 |      0 |
|   15    |     29 |      0 |      0 |      0 |
|   16    |      8 |      0 |      0 |      0 |
|   17    |     15 |      0 |      0 |      0 |
|   18    |     16 |      0 |      0 |      0 |
|   19    |    131 |      0 |      0 |      0 |
|   20    |    142 |      0 |      0 |      0 |
|   21    |    149 |      0 |      0 |      0 |
|   22    |    155 |      0 |      0 |      0 |
|   23    |    168 |      0 |      0 |      0 |
|   24    |    168 |      0 |      0 |      0 |
----------------------------------------------------------------------
|                    Optimization results                            |
----------------------------------------------------------------------
|                    Arc-Flow Clique Inequalities                    |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   24  |    KMeans |       Unlimited|  15346 | 12182 |     20.6%     |
----------------------------------------------------------------------
|                      Sailing costs                                 |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|   15346 |   2339 |   9398 |    752 |   2857 |        0   |      -0  |
----------------------------------------------------------------------
|                    Quantity discharged                             |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |    -10 |  -3900 |      0 |  -2900 |    -6810   |    No    |
|    2    |      0 |    400 |      0 |      0 |      400   |    No    |
|    3    |      0 |    500 |      0 |      0 |      500   |    No    |
|    4    |      0 |      0 |    500 |      0 |      500   |    No    |
|    5    |      0 |    400 |      0 |      0 |      400   |    No    |
|    6    |      0 |    500 |      0 |      0 |      500   |    No    |
|    7    |      0 |   1000 |      0 |      0 |     1000   |    No    |
|    8    |      0 |      0 |    800 |      0 |      800   |    No    |
|    9    |      0 |      0 |      0 |    800 |      800   |    No    |
|   10    |      0 |      0 |      0 |    600 |      600   |    No    |
|   11    |      0 |      0 |      0 |    500 |      500   |    No    |
|   12    |      0 |      0 |      0 |    100 |      100   |    No    |
|   13    |      0 |   1800 |      0 |      0 |     1800   |    No    |
|   14    |      0 |      0 |      0 |    500 |      500   |    No    |
|   15    |      0 |      0 |      0 |    300 |      300   |    No    |
|   16    |    300 |      0 |      0 |      0 |      300   |    No    |
|   17    |    700 |      0 |      0 |      0 |      700   |    No    |
|   18    |    900 |      0 |      0 |      0 |      900   |    No    |
|   19    |    300 |      0 |      0 |      0 |      300   |    No    |
|   20    |    700 |      0 |      0 |      0 |      700   |    No    |
|   21    |      0 |      0 |      0 |    400 |      400   |    No    |
|   22    |      0 |      0 |      0 |    800 |      800   |    No    |
|   23    |      0 |   1500 |      0 |      0 |     1500   |    No    |
|   24    |      0 |    800 |      0 |      0 |      800   |    No    |
Ship total|   2900 |   6900 |   1300 |   4000 |    15100   |          |
----------------------------------------------------------------------
|                      Time of visit                                 |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
----------------------------------------------------------------------
|    1    |      0 |     16 |     61 |     87 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |      3 |      0 |      0 |      0 |
|    4    |     22 |      0 |      0 |      0 |
|    5    |     30 |      0 |      0 |      0 |
|    6    |     35 |      0 |      0 |      0 |
|    7    |     42 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |     31 |      0 |      0 |      0 |
|   10    |     40 |      0 |      0 |      0 |
|   11    |      9 |      0 |      0 |      0 |
|   12    |      5 |      0 |      0 |      0 |
|   13    |     71 |      0 |      0 |      0 |
|   14    |      0 |      0 |      0 |      0 |
|   15    |    110 |      0 |      0 |      0 |
|   16    |      8 |      0 |      0 |      0 |
|   17    |     13 |      0 |      0 |      0 |
|   18    |     23 |      0 |      0 |      0 |
|   19    |     34 |      0 |      0 |      0 |
|   20    |     38 |      0 |      0 |      0 |
|   21    |    134 |      0 |      0 |      0 |
|   22    |    142 |      0 |      0 |      0 |
|   23    |    126 |      0 |      0 |      0 |
|   24    |    143 |      0 |      0 |      0 |
----------------------------------------------------------------------
|                    Optimization results                            |
----------------------------------------------------------------------
|                    Arc-Flow Time Windows                           |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|  24  |    KMeans |       Unlimited|  15510 | 12269 |     20.9%     |
----------------------------------------------------------------------
|                      Sailing costs                                 |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|   15510 |   2339 |   9205 |   2622 |   1344 |        0   |      -0  |
----------------------------------------------------------------------
|                    Quantity discharged                             |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |    -10 |  -3900 |  -1500 |  -1300 |    -6710   |    No    |
|    2    |      0 |    400 |      0 |      0 |     400    |    No    |
|    3    |      0 |    500 |      0 |      0 |     500    |    No    |
|    4    |      0 |    500 |      0 |      0 |     500    |    No    |
|    5    |      0 |    400 |      0 |      0 |     400    |    No    |
|    6    |      0 |      0 |      0 |    500 |     500    |    No    |
|    7    |      0 |   1000 |      0 |      0 |    1000    |    No    |
|    8    |      0 |      0 |    800 |      0 |     800    |    No    |
|    9    |      0 |      0 |      0 |    800 |     800    |    No    |
|   10    |      0 |      0 |    600 |      0 |     600    |    No    |
|   11    |      0 |      0 |      0 |    500 |     500    |    No    |
|   12    |      0 |      0 |      0 |    100 |     100    |    No    |
|   13    |      0 |   1800 |      0 |      0 |    1800    |    No    |
|   14    |      0 |      0 |      0 |    500 |     500    |    No    |
|   15    |      0 |      0 |    300 |      0 |     300    |    No    |
|   16    |    300 |      0 |      0 |      0 |     300    |    No    |
|   17    |    700 |      0 |      0 |      0 |     700    |    No    |
|   18    |    900 |      0 |      0 |      0 |     900    |    No    |
|   19    |    300 |      0 |      0 |      0 |     300    |    No    |
|   20    |    700 |      0 |      0 |      0 |     700    |    No    |
|   21    |      0 |      0 |    400 |      0 |     400    |    No    |
|   22    |      0 |    800 |      0 |      0 |     800    |    No    |
|   23    |      0 |   1500 |      0 |      0 |    1500    |    No    |
|   24    |      0 |      0 |    800 |      0 |     800    |    No    |
Ship total|   2900 |   6900 |   2900 |   2400 |    15100   |          |
----------------------------------------------------------------------
|                      Time of visit                                 |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
----------------------------------------------------------------------
|    1    |      0 |     64 |     70 |     76 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |      3 |      0 |      0 |      0 |
|    4    |     29 |      0 |      0 |      0 |
|    5    |     36 |      0 |      0 |      0 |
|    6    |    102 |      0 |      0 |      0 |
|    7    |     44 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |     85 |      0 |      0 |      0 |
|   10    |     11 |      0 |      0 |      0 |
|   11    |      9 |      0 |      0 |      0 |
|   12    |      5 |      0 |      0 |      0 |
|   13    |     73 |      0 |      0 |      0 |
|   14    |      0 |      0 |      0 |      0 |
|   15    |     89 |      0 |      0 |      0 |
|   16    |      8 |      0 |      0 |      0 |
|   17    |     13 |      0 |      0 |      0 |
|   18    |     23 |      0 |      0 |      0 |
|   19    |     34 |      0 |      0 |      0 |
|   20    |     38 |      0 |      0 |      0 |
|   21    |    113 |      0 |      0 |      0 |
|   22    |    125 |      0 |      0 |      0 |
|   23    |    138 |      0 |      0 |      0 |
|   24    |    131 |      0 |      0 |      0 |
-----------------------------------------------------------------------
|                    Optimization results                             |
-----------------------------------------------------------------------
|                    Arc-Flow Standard                                |
-----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   24   |    KMeans |       Unlimited|  15529 | 12234 |     21.2%     |
-----------------------------------------------------------------------
|                      Sailing costs                                  |
-----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
-----------------------------------------------------------------------
|   15529 |   3507 |   5645 |   1125 |    435 |        0   |    4816  |
-----------------------------------------------------------------------
|                    Quantity discharged                              |
-----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
-----------------------------------------------------------------------
|    1    |  -1810 |  -2000 |  -1300 |  -1400 |    -6510   |    No    |
|    2    |      0 |    400 |      0 |      0 |      400   |    No    |
|    3    |      0 |    500 |      0 |      0 |      500   |    No    |
|    4    |      0 |      0 |      0 |      0 |        0   |    No    |
|    5    |      0 |      0 |      0 |      0 |        0   |    No    |
|    6    |      0 |      0 |      0 |      0 |        0   |    No    |
|    7    |      0 |   1000 |      0 |      0 |     1000   |    No    |
|    8    |      0 |      0 |    800 |      0 |      800   |    No    |
|    9    |      0 |      0 |      0 |      0 |        0   |    No    |
|   10    |      0 |      0 |    600 |      0 |      600   |    No    |
|   11    |      0 |      0 |      0 |    500 |      500   |    No    |
|   12    |      0 |      0 |      0 |    100 |      100   |    No    |
|   13    |      0 |      0 |      0 |      0 |        0   |    No    |
|   14    |      0 |      0 |      0 |    500 |      500   |    No    |
|   15    |      0 |      0 |    300 |      0 |      300   |    No    |
|   16    |      0 |      0 |    300 |      0 |      300   |    No    |
|   17    |      0 |    700 |      0 |      0 |      700   |    No    |
|   18    |      0 |    900 |      0 |      0 |      900   |    No    |
|   19    |    300 |      0 |      0 |      0 |      300   |    No    |
|   20    |      0 |    700 |      0 |      0 |      700   |    No    |
|   21    |    400 |      0 |      0 |      0 |      400   |    No    |
|   22    |    800 |      0 |      0 |      0 |      800   |    No    |
|   23    |   1500 |      0 |      0 |      0 |     1500   |    No    |
|   24    |      0 |      0 |      0 |      0 |        0   |    No    |
Ship total|   3000 |   4200 |   2000 |   1100 |    10300   |          |
-----------------------------------------------------------------------
|                      Time of visit                                  |
-----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
-----------------------------------------------------------------------
|    1    |      0 |     16 |     22 |     27 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |      3 |      0 |      0 |      0 |
|    4    |     55 |      0 |      0 |      0 |
|    5    |     48 |      0 |      0 |      0 |
|    6    |     42 |      0 |      0 |      0 |
|    7    |     36 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |     45 |      0 |      0 |      0 |
|   10    |     11 |      0 |      0 |      0 |
|   11    |      9 |      0 |      0 |      0 |
|   12    |      5 |      0 |      0 |      0 |
|   13    |      3 |      0 |      0 |      0 |
|   14    |      0 |      0 |      0 |      0 |
|   15    |     35 |      0 |      0 |      0 |
|   16    |     41 |      0 |      0 |      0 |
|   17    |     74 |      0 |      0 |      0 |
|   18    |     85 |      0 |      0 |      0 |
|   19    |     52 |      0 |      0 |      0 |
|   20    |     97 |      0 |      0 |      0 |
|   21    |     59 |      0 |      0 |      0 |
|   22    |     65 |      0 |      0 |      0 |
|   23    |     78 |      0 |      0 |      0 |
|   24    |     76 |      0 |      0 |      0 |
-----------------------------------------------------------------------
|                    Optimization results                             |
-----------------------------------------------------------------------
|                    Arc-Flow Sub-Tour Elimination                    |
-----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   24   |    KMeans |       Unlimited|  15578 | 12210 |     21.6%     |
-----------------------------------------------------------------------
|                      Sailing costs                                  |
-----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
-----------------------------------------------------------------------
|   15578 |   2955 |   9169 |   2102 |   1353 |        0   |       0  |
-----------------------------------------------------------------------
|                    Quantity discharged                              |
-----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
-----------------------------------------------------------------------
|    1    |  -1410 |  -2600 |  -1500 |  -1500 |    -7010   |    No    |
|    2    |      0 |    400 |      0 |      0 |      400   |    No    |
|    3    |      0 |    500 |      0 |      0 |      500   |    No    |
|    4    |      0 |    500 |      0 |      0 |      500   |    No    |
|    5    |      0 |    571 |      0 |      0 |      571   |    No    |
|    6    |      0 |      0 |      0 |    500 |      500   |    No    |
|    7    |      0 |      0 |      0 |   1000 |     1000   |    No    |
|    8    |      0 |      0 |    900 |      0 |      900   |    No    |
|    9    |      0 |    800 |      0 |      0 |      800   |    No    |
|   10    |      0 |      0 |    600 |      0 |      600   |    No    |
|   11    |      0 |      0 |      0 |    500 |      500   |    No    |
|   12    |      0 |      0 |      0 |    100 |      100   |    No    |
|   13    |   1800 |      0 |      0 |      0 |     1800   |    No    |
|   14    |      0 |      0 |      0 |    608 |      608   |    No    |
|   15    |      0 |    300 |      0 |      0 |      300   |    No    |
|   16    |    300 |      0 |      0 |      0 |      300   |    No    |
|   17    |    700 |      0 |      0 |      0 |      700   |    No    |
|   18    |    900 |      0 |      0 |      0 |      900   |    No    |
|   19    |      0 |      0 |    300 |      0 |      300   |    No    |
|   20    |    700 |      0 |      0 |      0 |      700   |    No    |
|   21    |      0 |      0 |    400 |      0 |      400   |    No    |
|   22    |      0 |      0 |    800 |      0 |      800   |    No    |
|   23    |      0 |   1500 |      0 |      0 |     1500   |    No    |
|   24    |      0 |    800 |      0 |      0 |      800   |    No    |
Ship total|   4400 |   5371 |   3000 |   2708 |    15480   |          |
-----------------------------------------------------------------------
|                      Time of visit                                  |
-----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
-----------------------------------------------------------------------
|    1    |      0 |     51 |     57 |     63 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |      3 |      0 |      0 |      0 |
|    4    |     29 |      0 |      0 |      0 |
|    5    |     36 |      0 |      0 |      0 |
|    6    |     85 |      0 |      0 |      0 |
|    7    |     72 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |     50 |      0 |      0 |      0 |
|   10    |     46 |      0 |      0 |      0 |
|   11    |      9 |      0 |      0 |      0 |
|   12    |      5 |      0 |      0 |      0 |
|   13    |      3 |      0 |      0 |      0 |
|   14    |      0 |      0 |      0 |      0 |
|   15    |     79 |      0 |      0 |      0 |
|   16    |     87 |      0 |      0 |      0 |
|   17    |     91 |      0 |      0 |      0 |
|   18    |    102 |      0 |      0 |      0 |
|   19    |     86 |      0 |      0 |      0 |
|   20    |    114 |      0 |      0 |      0 |
|   21    |     94 |      0 |      0 |      0 |
|   22    |    102 |      0 |      0 |      0 |
|   23    |    105 |      0 |      0 |      0 |
|   24    |    122 |      0 |      0 |      0 |
```

Compatibility Scenarios

```
----------------------------------------------------------------------
|                   Optimization results                             |
----------------------------------------------------------------------
|          Arc-Flow Standard - Partly Limited Compatibility          |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   12  |     Zones |  Partly Limited|  13340 | 13340 |     0.0%      |
----------------------------------------------------------------------
|                      Sailing costs                                 |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|   13340 |   2879 |   8313 |   1242 |    906 |        0   |       0  |
----------------------------------------------------------------------
|                    Quantity discharged                             |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |    -10 |  -2300 |  -2900 |  -1210 |    -6420   |    No    |
|    2    |      0 |     10 |      0 |      0 |       10   |    No    |
|    3    |      0 |    900 |      0 |      0 |      900   |    No    |
|    4    |      0 |      0 |      0 |      0 |        0   |    No    |
|    5    |      0 |    500 |      0 |      0 |      500   |    No    |
|    6    |      0 |   1590 |      0 |   1110 |     2700   |    No    |
|    7    |      0 |      0 |   1500 |    100 |     1600   |    No    |
|    8    |      0 |      0 |   1500 |   1500 |     3000   |    No    |
|    9    |      0 |      0 |   1400 |      0 |     1400   |    No    |
|   10    |   1900 |      0 |      0 |      0 |     1900   |    No    |
|   11    |   1100 |      0 |      0 |      0 |     1100   |    No    |
|   12    |      0 |   2300 |      0 |      0 |     2300   |    No    |
Ship total|   3000 |   5300 |   4400 |   2710 |    15410   |          |
----------------------------------------------------------------------
|                      Time of visit                                 |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4| Visit 5|
----------------------------------------------------------------------
|    1    |      0 |     23 |     33 |     65 |    124 |
|    2    |      0 |      0 |      0 |      0 |      0 |
|    3    |      1 |      0 |      0 |      0 |      0 |
|    4    |      0 |      0 |      0 |      0 |      0 |
|    5    |     30 |      0 |      0 |      0 |      0 |
|    6    |     65 |     69 |      0 |      0 |      0 |
|    7    |      0 |     46 |      0 |      0 |      0 |
|    8    |      0 |     33 |      0 |      0 |      0 |
|    9    |     85 |      0 |      0 |      0 |      0 |
|   10    |     19 |      0 |      0 |      0 |      0 |
|   11    |    168 |      0 |      0 |      0 |      0 |
|   12    |    163 |      0 |      0 |      0 |      0 |
----------------------------------------------------------------------
|                   Optimization results                             |
----------------------------------------------------------------------
|             Arc-Flow Standard - Limited Compatibility              |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|  12  |     Zones |         Limited|  14734 | 14734 |     0.0%      |
----------------------------------------------------------------------
|                     Sailing costs                                  |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|   14734 |   8930 |   3656 |   1401 |    747 |        0   |       0  |
----------------------------------------------------------------------
|                   Quantity discharged                              |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |  -2310 |      0 |  -2710 |  -1400 |    -6420   |    No    |
|    2    |      0 |     10 |      0 |      0 |       10   |    No    |
|    3    |      0 |    900 |      0 |      0 |      900   |    No    |
|    4    |      0 |      0 |      0 |      0 |        0   |    No    |
|    5    |      0 |    500 |      0 |      0 |      500   |    No    |
|    6    |      0 |   1590 |   1110 |      0 |     2700   |    No    |
|    7    |      0 |      0 |   1600 |      0 |     1600   |    No    |
|    8    |      0 |      0 |   1500 |   1500 |     3000   |    No    |
|    9    |      0 |      0 |      0 |   1400 |     1400   |    No    |
|   10    |   1900 |      0 |      0 |      0 |     1900   |    No    |
|   11    |    800 |      0 |      0 |      0 |      800   |    No    |
|   12    |   2300 |      0 |      0 |      0 |     2300   |    No    |
Ship total|   5000 |   3000 |   4210 |   2900 |    15110   |          |
----------------------------------------------------------------------
|                     Time of visit                                  |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4| Visit 5|
----------------------------------------------------------------------
|    1    |      0 |     81 |     87 |    136 |    141 |
|    2    |      0 |      0 |      0 |      0 |      0 |
|    3    |      1 |      0 |      0 |      0 |      0 |
|    4    |      0 |      0 |      0 |      0 |      0 |
|    5    |     30 |      0 |      0 |      0 |      0 |
|    6    |    162 |    168 |      0 |      0 |      0 |
|    7    |      0 |    149 |      0 |      0 |      0 |
|    8    |      0 |    103 |      0 |      0 |      0 |
|    9    |    168 |      0 |      0 |      0 |      0 |
|   10    |     19 |      0 |      0 |      0 |      0 |
|   11    |     54 |      0 |      0 |      0 |      0 |
|   12    |    168 |      0 |      0 |      0 |      0 |
-----------------------------------------------------------------------
|                    Optimization results                             |
-----------------------------------------------------------------------
|           Arc-Flow Standard - Partly Limited Compatibility          |
-----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   19  |   Natural |  Partly Limited|  20641 | 20641 |     0.0%      |
-----------------------------------------------------------------------
|                      Sailing costs                                  |
-----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
-----------------------------------------------------------------------
|   20641 |   7687 |   1619 |    990 |   1145 |     2300   |    6900  |
-----------------------------------------------------------------------
|                    Quantity discharged                              |
-----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
-----------------------------------------------------------------------
|    1    |  -2300 |      0 |   -800 |  -1000 |    -4100   |    No    |
|    2    |    900 |      0 |      0 |      0 |      900   |    No    |
|    3    |    500 |      0 |      0 |      0 |      500   |    No    |
|    4    |    400 |      0 |      0 |      0 |      400   |    No    |
|    5    |    500 |      0 |      0 |      0 |      500   |    No    |
|    6    |      0 |      0 |   1000 |      0 |     1000   |    No    |
|    7    |      0 |      0 |      0 |    800 |      800   |    No    |
|    8    |      0 |      0 |      0 |    300 |      300   |    No    |
|    9    |      0 |      0 |   1300 |      0 |     1300   |    No    |
|   10    |      0 |      0 |      0 |    400 |      400   |    No    |
|   11    |      0 |      0 |      0 |      0 |        0   |    Yes   |
|   12    |      0 |      0 |      0 |    600 |      600   |    No    |
|   13    |      0 |      0 |      0 |    400 |      400   |    No    |
|   14    |   1300 |      0 |      0 |      0 |     1300   |    No    |
|   15    |    900 |      0 |      0 |      0 |      900   |    No    |
|   16    |      0 |    400 |      0 |      0 |      400   |    No    |
|   17    |    800 |      0 |      0 |      0 |      800   |    No    |
|   18    |      0 |   1500 |      0 |      0 |     1500   |    No    |
|   19    |      0 |    800 |      0 |      0 |      800   |    No    |
Ship total|   5300 |   2700 |   2300 |   2500 |    12800   |          |
-----------------------------------------------------------------------
|                      Time of visit                                  |
-----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
-----------------------------------------------------------------------
|    1    |     26 |     59 |     59 |     68 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |     28 |      0 |      0 |      0 |
|    4    |     35 |      0 |      0 |      0 |
|    5    |     41 |      0 |      0 |      0 |
|    6    |     87 |      0 |      0 |      0 |
|    7    |      4 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |      0 |      0 |      0 |      0 |
|   10    |     20 |      0 |      0 |      0 |
|   11    |      0 |      0 |      0 |      0 |
|   12    |     39 |      0 |      0 |      0 |
|   13    |     48 |      0 |      0 |      0 |
|   14    |     82 |      0 |      0 |      0 |
|   15    |     99 |      0 |      0 |      0 |
|   16    |     29 |      0 |      0 |      0 |
|   17    |    168 |      0 |      0 |      0 |
|   18    |     11 |      0 |      0 |      0 |
|   19    |      0 |      0 |      0 |      0 |
----------------------------------------------------------------------
|                     Optimization results                           |
----------------------------------------------------------------------
|               Arc-Flow Standard - Limited Compatibility            |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   19  |   Natural |         Limited|  21422 | 21422 |     0.0%      |
----------------------------------------------------------------------
|                       Sailing costs                                |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|   21422 |   2888 |   6744 |   1065 |   1525 |     2300   |    6900  |
----------------------------------------------------------------------
|                     Quantity discharged                            |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |      0 |  -2700 |  -1200 |  -1500 |    -5400   |    No    |
|    2    |    900 |      0 |      0 |      0 |      900   |    No    |
|    3    |    500 |      0 |      0 |      0 |      500   |    No    |
|    4    |    400 |      0 |      0 |      0 |      400   |    No    |
|    5    |      0 |      0 |      0 |    500 |      500   |    No    |
|    6    |      0 |      0 |      0 |   1000 |     1000   |    No    |
|    7    |      0 |      0 |      0 |    800 |      800   |    No    |
|    8    |      0 |      0 |      0 |    300 |      300   |    No    |
|    9    |      0 |      0 |   1300 |      0 |     1300   |    No    |
|   10    |      0 |      0 |    400 |      0 |      400   |    No    |
|   11    |      0 |      0 |      0 |      0 |        0   |    Yes   |
|   12    |      0 |      0 |    600 |      0 |      600   |    No    |
|   13    |      0 |      0 |    400 |      0 |      400   |    No    |
|   14    |      0 |   1300 |      0 |      0 |     1300   |    No    |
|   15    |      0 |    900 |      0 |      0 |      900   |    No    |
|   16    |      0 |    400 |      0 |      0 |      400   |    No    |
|   17    |      0 |    800 |      0 |      0 |      800   |    No    |
|   18    |      0 |   1500 |      0 |      0 |     1500   |    No    |
|   19    |      0 |    800 |      0 |      0 |      800   |    No    |
Ship total|   1800 |   5700 |   2700 |   2600 |    12800   |          |
----------------------------------------------------------------------
|                       Time of visit                                |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
----------------------------------------------------------------------
|    1    |     53 |     64 |    137 |    137 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |     28 |      0 |      0 |      0 |
|    4    |    168 |      0 |      0 |      0 |
|    5    |    168 |      0 |      0 |      0 |
|    6    |    155 |      0 |      0 |      0 |
|    7    |      4 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |      0 |      0 |      0 |      0 |
|   10    |    144 |      0 |      0 |      0 |
|   11    |      0 |      0 |      0 |      0 |
|   12    |    159 |      0 |      0 |      0 |
|   13    |    168 |      0 |      0 |      0 |
|   14    |    135 |      0 |      0 |      0 |
|   15    |    152 |      0 |      0 |      0 |
|   16    |     29 |      0 |      0 |      0 |
|   17    |    168 |      0 |      0 |      0 |
|   18    |     11 |      0 |      0 |      0 |
|   19    |      0 |      0 |      0 |      0 |
----------------------------------------------------------------------
|                    Optimization results                            |
----------------------------------------------------------------------
|      Arc-Flow Standard - Partly Limited Compatibility              |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   24  |    KMeans |  Partly Limited|  22287 | 22287 |     0.0%      |
----------------------------------------------------------------------
|                      Sailing costs                                 |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|   22287 |   3996 |   8172 |   1318 |   1600 |     1800   |    5400  |
----------------------------------------------------------------------
|                    Quantity discharged                             |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |    -10 |  -1700 |  -1300 |  -1500 |    -4510   |    No    |
|    2    |      0 |    400 |      0 |      0 |      400   |    No    |
|    3    |      0 |    500 |      0 |      0 |      500   |    No    |
|    4    |      0 |    500 |      0 |      0 |      500   |    No    |
|    5    |      0 |    400 |      0 |      0 |      400   |    No    |
|    6    |      0 |    500 |      0 |      0 |      500   |    No    |
|    7    |      0 |      0 |      0 |   1000 |     1000   |    No    |
|    8    |      0 |      0 |    800 |      0 |      800   |    No    |
|    9    |      0 |      0 |      0 |    800 |      800   |    No    |
|   10    |      0 |      0 |    600 |      0 |      600   |    No    |
|   11    |      0 |      0 |      0 |    500 |      500   |    No    |
|   12    |      0 |      0 |      0 |    100 |      100   |    No    |
|   13    |      0 |      0 |      0 |      0 |        0   |    Yes   |
|   14    |      0 |      0 |      0 |    500 |      500   |    No    |
|   15    |      0 |      0 |    300 |      0 |      300   |    No    |
|   16    |      0 |      0 |    300 |      0 |      300   |    No    |
|   17    |      0 |      0 |    700 |      0 |      700   |    No    |
|   18    |      0 |    900 |      0 |      0 |      900   |    No    |
|   19    |    300 |      0 |      0 |      0 |      300   |    No    |
|   20    |      0 |    700 |      0 |      0 |      700   |    No    |
|   21    |    400 |      0 |      0 |      0 |      400   |    No    |
|   22    |      0 |    800 |      0 |      0 |      800   |    No    |
|   23    |   1500 |      0 |      0 |      0 |     1500   |    No    |
|   24    |    800 |      0 |      0 |      0 |      800   |    No    |
Ship total|   3000 |   4700 |   2700 |   2900 |    13300   |          |
----------------------------------------------------------------------
|                      Time of visit                                 |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
----------------------------------------------------------------------
|    1    |      0 |    111 |    111 |    116 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |      3 |      0 |      0 |      0 |
|    4    |     29 |      0 |      0 |      0 |
|    5    |     36 |      0 |      0 |      0 |
|    6    |     41 |      0 |      0 |      0 |
|    7    |    141 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |     14 |      0 |      0 |      0 |
|   10    |     11 |      0 |      0 |      0 |
|   11    |    126 |      0 |      0 |      0 |
|   12    |      5 |      0 |      0 |      0 |
|   13    |      0 |      0 |      0 |      0 |
|   14    |      0 |      0 |      0 |      0 |
|   15    |    124 |      0 |      0 |      0 |
|   16    |    130 |      0 |      0 |      0 |
|   17    |    135 |      0 |      0 |      0 |
|   18    |    143 |      0 |      0 |      0 |
|   19    |     18 |      0 |      0 |      0 |
|   20    |    155 |      0 |      0 |      0 |
|   21    |     25 |      0 |      0 |      0 |
|   22    |    168 |      0 |      0 |      0 |
|   23    |     34 |      0 |      0 |      0 |
|   24    |     52 |      0 |      0 |      0 |
----------------------------------------------------------------------
|                   Optimization results                             |
----------------------------------------------------------------------
|             Arc-Flow Standard - Limited Compatibility              |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|  24  |    KMeans |         Limited|  24319 | 24319 |     0.0%      |
----------------------------------------------------------------------
|                     Sailing costs                                  |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|   24319 |   9807 |   3868 |   1035 |    810 |     2200   |    6600  |
----------------------------------------------------------------------
|                   Quantity discharged                              |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |  -2710 |      0 |  -1400 |  -1300 |    -5410   |    No    |
|    2    |      0 |    400 |      0 |      0 |      400   |    No    |
|    3    |      0 |    500 |      0 |      0 |      500   |    No    |
|    4    |      0 |    500 |      0 |      0 |      500   |    No    |
|    5    |      0 |     10 |      0 |      0 |       10   |    Yes   |
|    6    |      0 |    500 |      0 |      0 |      500   |    No    |
|    7    |      0 |   1000 |      0 |      0 |     1000   |    No    |
|    8    |      0 |      0 |    800 |      0 |      800   |    No    |
|    9    |      0 |      0 |    800 |      0 |      800   |    No    |
|   10    |      0 |      0 |    600 |      0 |      600   |    No    |
|   11    |      0 |      0 |    500 |      0 |      500   |    No    |
|   12    |      0 |      0 |    100 |      0 |      100   |    No    |
|   13    |      0 |      0 |      0 |      0 |        0   |    Yes   |
|   14    |      0 |      0 |      0 |    500 |      500   |    No    |
|   15    |      0 |      0 |      0 |    300 |      300   |    No    |
|   16    |      0 |      0 |      0 |    300 |      300   |    No    |
|   17    |      0 |      0 |      0 |    700 |      700   |    No    |
|   18    |    900 |      0 |      0 |      0 |      900   |    No    |
|   19    |    300 |      0 |      0 |      0 |      300   |    No    |
|   20    |    700 |      0 |      0 |      0 |      700   |    No    |
|   21    |    400 |      0 |      0 |      0 |      400   |    No    |
|   22    |    800 |      0 |      0 |      0 |      800   |    No    |
|   23    |   1500 |      0 |      0 |      0 |     1500   |    No    |
|   24    |    800 |      0 |      0 |      0 |      800   |    No    |
Ship total|   5400 |   2910 |   2800 |   1800 |    12910   |          |
----------------------------------------------------------------------
|                     Time of visit                                  |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
----------------------------------------------------------------------
|    1    |      0 |      4 |     22 |     75 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |    125 |      0 |      0 |      0 |
|    4    |    151 |      0 |      0 |      0 |
|    5    |    158 |      0 |      0 |      0 |
|    6    |    161 |      0 |      0 |      0 |
|    7    |    168 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |     41 |      0 |      0 |      0 |
|   10    |     11 |      0 |      0 |      0 |
|   11    |     33 |      0 |      0 |      0 |
|   12    |     29 |      0 |      0 |      0 |
|   13    |      0 |      0 |      0 |      0 |
|   14    |      0 |      0 |      0 |      0 |
|   15    |     17 |      0 |      0 |      0 |
|   16    |     23 |      0 |      0 |      0 |
|   17    |     29 |      0 |      0 |      0 |
|   18    |     14 |      0 |      0 |      0 |
|   19    |     54 |      0 |      0 |      0 |
|   20    |     26 |      0 |      0 |      0 |
|   21    |    141 |      0 |      0 |      0 |
|   22    |     39 |      0 |      0 |      0 |
|   23    |    150 |      0 |      0 |      0 |
|   24    |    168 |      0 |      0 |      0 |
```

Consumption Scenarios

```
-----------------------------------------------------------------------
|                    Optimization results                             |
-----------------------------------------------------------------------
|                   Arc-Flow Standard Low Consumption                 |
-----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   12  |     Zones |       Unlimited|   6788 |  6788 |     0.0%      |
-----------------------------------------------------------------------
|                      Sailing costs                                  |
-----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
-----------------------------------------------------------------------
|    6788 |   3647 |    108 |    941 |   2091 |        0   |      -0  |
-----------------------------------------------------------------------
|                    Quantity discharged                              |
-----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
-----------------------------------------------------------------------
|    1    |    -10 |     -0 |     -0 |  -2050 |    -2060   |    No    |
|    2    |      0 |     10 |      0 |      0 |       10   |    No    |
|    3    |      0 |    450 |      0 |      0 |      450   |    No    |
|    4    |      0 |      0 |      0 |      0 |        0   |    No    |
|    5    |      0 |      0 |    700 |      0 |      700   |    No    |
|    6    |      0 |      0 |      0 |   1350 |     1350   |    No    |
|    7    |      0 |      0 |    800 |      0 |      800   |    No    |
|    8    |      0 |      0 |      0 |   1500 |     1500   |    No    |
|    9    |      0 |      0 |      0 |    700 |      700   |    No    |
|   10    |    950 |      0 |      0 |      0 |      950   |    No    |
|   11    |    400 |      0 |      0 |      0 |      400   |    No    |
|   12    |   1150 |      0 |      0 |      0 |     1150   |    No    |
Ship total|   2500 |    460 |   1500 |   3550 |     8010   |          |
-----------------------------------------------------------------------
|                      Time of visit                                  |
-----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
-----------------------------------------------------------------------
|    1    |      0 |     96 |    141 |    146 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |      1 |      0 |      0 |      0 |
|    4    |      0 |      0 |      0 |      0 |
|    5    |     43 |      0 |      0 |      0 |
|    6    |    163 |      0 |      0 |      0 |
|    7    |      0 |     12 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |    113 |      0 |      0 |      0 |
|   10    |     19 |      0 |      0 |      0 |
|   11    |     50 |      0 |      0 |      0 |
|   12    |     63 |      0 |      0 |      0 |
-----------------------------------------------------------------------
|                    Optimization results                             |
-----------------------------------------------------------------------
|                Arc-Flow Standard High Consumption                   |
-----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   12  |     Zones |       Unlimited|  23603 | 23603 |     0.0%      |
-----------------------------------------------------------------------
|                      Sailing costs                                  |
-----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
-----------------------------------------------------------------------
|   23603 |   5508 |   6677 |      0 |   1618 |     4200   |    5600  |
-----------------------------------------------------------------------
|                    Quantity discharged                              |
-----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
-----------------------------------------------------------------------
|    1    |  -10085 |  -3000|      0 |  -1200 |    -14285  |    No    |
|    2    |       0 |     10 |      0 |      0 |      10    |    No    |
|    3    |       0 |   1350 |      0 |      0 |    1350    |    No    |
|    4    |       0 |      0 |      0 |      0 |       0    |    No    |
|    5    |       0 |      0 |      0 |      0 |       0    |    Yes   |
|    6    |    2410 |   1640 |      0 |      0 |    4050    |    No    |
|    7    |     900 |      0 |   1500 |      0 |    2400    |    No    |
|    8    |    3000 |      0 |      0 |   1500 |    4500    |    No    |
|    9    |    2100 |      0 |      0 |      0 |    2100    |    No    |
|   10    |       0 |   3000 |      0 |      0 |    3000    |    No    |
|   11    |       0 |      0 |      0 |   1200 |    1200    |    No    |
|   12    |       0 |      0 |      0 |      0 |       0    |    Yes   |
Ship total|    8410 |   6000 |   1500 |   2700 |   18610    |          |
-----------------------------------------------------------------------
|                      Time of visit                                  |
-----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
-----------------------------------------------------------------------
|    1    |       0 |     84 |     96 |    101 |
|    2    |       0 |      0 |      0 |      0 |
|    3    |       1 |      0 |      0 |      0 |
|    4    |       0 |      0 |      0 |      0 |
|    5    |       0 |      0 |      0 |      0 |
|    6    |      37 |    168 |      0 |      0 |
|    7    |       0 |     25 |      0 |      0 |
|    8    |       0 |     99 |      0 |      0 |
|    9    |      54 |      0 |      0 |      0 |
|   10    |     134 |      0 |      0 |      0 |
|   11    |     134 |      0 |      0 |      0 |
|   12    |       0 |      0 |      0 |      0 |
-----------------------------------------------------------------------
|                    Optimization results                             |
-----------------------------------------------------------------------
|                 Arc-Flow Standard - Low Consumption                 |
-----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|  19  |   Natural |       Unlimited|   5039 |  5039 |     0.0%      |
-----------------------------------------------------------------------
|                      Sailing costs                                  |
-----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
-----------------------------------------------------------------------
|    5039 |      0 |   2245 |   1874 |    919 |        0   |      -0  |
-----------------------------------------------------------------------
|                    Quantity discharged                              |
-----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
-----------------------------------------------------------------------
|    1    |      0 |      0 |  -2300 |      0 |    -2300   |    No    |
|    2    |    450 |      0 |      0 |      0 |      450   |    No    |
|    3    |      0 |      0 |      0 |    250 |      250   |    No    |
|    4    |      0 |      0 |      0 |    200 |      200   |    No    |
|    5    |      0 |      0 |      0 |    250 |      250   |    No    |
|    6    |      0 |      0 |      0 |    500 |      500   |    No    |
|    7    |      0 |      0 |    400 |      0 |      400   |    No    |
|    8    |      0 |      0 |      0 |    150 |      150   |    No    |
|    9    |      0 |      0 |    650 |      0 |      650   |    No    |
|   10    |      0 |      0 |    450 |      0 |      450   |    No    |
|   11    |      0 |      0 |   1150 |      0 |     1150   |    No    |
|   12    |      0 |      0 |    300 |      0 |      300   |    No    |
|   13    |      0 |      0 |    200 |      0 |      200   |    No    |
|   14    |      0 |      0 |    650 |      0 |      650   |    No    |
|   15    |      0 |    450 |      0 |      0 |      450   |    No    |
|   16    |      0 |    200 |      0 |      0 |      200   |    No    |
|   17    |      0 |    400 |      0 |      0 |      400   |    No    |
|   18    |      0 |    750 |      0 |      0 |      750   |    No    |
|   19    |      0 |    400 |      0 |      0 |      400   |    No    |
Ship total|    450 |   2200 |   3800 |   1350 |     7800   |          |
-----------------------------------------------------------------------
|                      Time of visit                                  |
-----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
-----------------------------------------------------------------------
|    1    |     36 |     64 |      0 |      0 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |     32 |      0 |      0 |      0 |
|    4    |     25 |      0 |      0 |      0 |
|    5    |     18 |      0 |      0 |      0 |
|    6    |      7 |      0 |      0 |      0 |
|    7    |     16 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |      0 |      0 |      0 |      0 |
|   10    |     30 |      0 |      0 |      0 |
|   11    |     45 |      0 |      0 |      0 |
|   12    |     78 |      0 |      0 |      0 |
|   13    |     85 |      0 |      0 |      0 |
|   14    |     92 |      0 |      0 |      0 |
|   15    |     38 |      0 |      0 |      0 |
|   16    |     32 |      0 |      0 |      0 |
|   17    |     22 |      0 |      0 |      0 |
|   18    |     10 |      0 |      0 |      0 |
|   19    |      0 |      0 |      0 |      0 |
----------------------------------------------------------------------
|                    Optimization results                            |
----------------------------------------------------------------------
|              Arc-Flow Standard - High Consumption                  |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   19  |   Natural |       Unlimited|  32579 | 28005 |     14.0%     |
----------------------------------------------------------------------
|                      Sailing costs                                 |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|   32579 |      0 |   5934 |   2020 |   3275 |     9150   |   12200  |
----------------------------------------------------------------------
|                    Quantity discharged                             |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |      0 |  -2720 |  -1210 |  -2700 |    -6630   |    No    |
|    2    |   1350 |      0 |      0 |      0 |     1350   |    No    |
|    3    |      0 |      0 |    750 |      0 |      750   |    No    |
|    4    |      0 |      0 |      0 |      0 |        0   |    Yes   |
|    5    |      0 |      0 |    750 |      0 |      750   |    No    |
|    6    |      0 |      0 |      0 |   1500 |     1500   |    No    |
|    7    |      0 |      0 |   1200 |      0 |     1200   |    No    |
|    8    |      0 |      0 |      0 |    450 |      450   |    No    |
|    9    |      0 |      0 |     10 |      0 |       10   |    Yes   |
|   10    |      0 |      0 |      0 |    600 |      600   |    No    |
|   11    |      0 |      0 |      0 |      0 |        0   |    Yes   |
|   12    |      0 |    900 |      0 |      0 |      900   |    No    |
|   13    |      0 |    600 |      0 |      0 |      600   |    No    |
|   14    |      0 |     10 |      0 |      0 |       10   |    Yes   |
|   15    |      0 |   1350 |      0 |      0 |     1350   |    No    |
|   16    |      0 |    600 |      0 |      0 |      600   |    No    |
|   17    |      0 |      0 |      0 |   1200 |     1200   |    No    |
|   18    |      0 |   2250 |      0 |      0 |     2250   |    No    |
|   19    |      0 |     10 |      0 |      0 |       10   |    Yes   |
Ship total|   1350 |   5720 |   2710 |   3750 |    13530   |          |
----------------------------------------------------------------------
|                      Time of visit                                 |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
----------------------------------------------------------------------
|    1    |     16 |     34 |     54 |     65 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |     72 |      0 |      0 |      0 |
|    4    |      0 |      0 |      0 |      0 |
|    5    |     58 |      0 |      0 |      0 |
|    6    |     38 |      0 |      0 |      0 |
|    7    |     13 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |      0 |      0 |      0 |      0 |
|   10    |      9 |      0 |      0 |      0 |
|   11    |      0 |      0 |      0 |      0 |
|   12    |     72 |      0 |      0 |      0 |
|   13    |     81 |      0 |      0 |      0 |
|   14    |     88 |      0 |      0 |      0 |
|   15    |    101 |      0 |      0 |      0 |
|   16    |     29 |      0 |      0 |      0 |
|   17    |    102 |      0 |      0 |      0 |
|   18    |      8 |      0 |      0 |      0 |
|   19    |      0 |      0 |      0 |      0 |
----------------------------------------------------------------------
|                   Optimization results                             |
----------------------------------------------------------------------
|              Arc-Flow Standard - Low Consumption                   |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   24  |    KMeans |       Unlimited|   7454 |  7454 |     0.0%      |
----------------------------------------------------------------------
|                     Sailing costs                                  |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|    7454 |   4758 |    145 |   1973 |    578 |        0   |      -0  |
----------------------------------------------------------------------
|                   Quantity discharged                              |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |  -3010 |      0 |  -1500 |     -0 |    -4510   |    No    |
|    2    |      0 |    200 |      0 |      0 |      200   |    No    |
|    3    |      0 |    250 |      0 |      0 |      250   |    No    |
|    4    |      0 |      0 |    250 |      0 |      250   |    No    |
|    5    |      0 |      0 |    200 |      0 |      200   |    No    |
|    6    |      0 |      0 |    250 |      0 |      250   |    No    |
|    7    |      0 |      0 |    500 |      0 |      500   |    No    |
|    8    |      0 |      0 |    400 |      0 |      400   |    No    |
|    9    |      0 |      0 |    400 |      0 |      400   |    No    |
|   10    |      0 |      0 |    300 |      0 |      300   |    No    |
|   11    |      0 |      0 |    650 |      0 |      650   |    No    |
|   12    |      0 |      0 |     50 |      0 |       50   |    No    |
|   13    |      0 |      0 |      0 |    900 |      900   |    No    |
|   14    |      0 |      0 |      0 |    250 |      250   |    No    |
|   15    |    150 |      0 |      0 |      0 |      150   |    No    |
|   16    |    150 |      0 |      0 |      0 |      150   |    No    |
|   17    |      0 |      0 |      0 |    350 |      350   |    No    |
|   18    |    450 |      0 |      0 |      0 |      450   |    No    |
|   19    |    150 |      0 |      0 |      0 |      150   |    No    |
|   20    |    350 |      0 |      0 |      0 |      350   |    No    |
|   21    |    200 |      0 |      0 |      0 |      200   |    No    |
|   22    |    400 |      0 |      0 |      0 |      400   |    No    |
|   23    |    750 |      0 |      0 |      0 |      750   |    No    |
|   24    |    400 |      0 |      0 |      0 |      400   |    No    |
Ship total|   3000 |    450 |   3000 |   1500 |     7950   |          |
----------------------------------------------------------------------
|                     Time of visit                                  |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
----------------------------------------------------------------------
|    1    |      0 |      0 |      0 |    120 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |      3 |      0 |      0 |      0 |
|    4    |    168 |      0 |      0 |      0 |
|    5    |    162 |      0 |      0 |      0 |
|    6    |    155 |      0 |      0 |      0 |
|    7    |    145 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |     20 |      0 |      0 |      0 |
|   10    |    133 |      0 |      0 |      0 |
|   11    |     30 |      0 |      0 |      0 |
|   12    |     37 |      0 |      0 |      0 |
|   13    |      3 |      0 |      0 |      0 |
|   14    |      0 |      0 |      0 |      0 |
|   15    |     18 |      0 |      0 |      0 |
|   16    |     22 |      0 |      0 |      0 |
|   17    |    168 |      0 |      0 |      0 |
|   18    |     30 |      0 |      0 |      0 |
|   19    |     39 |      0 |      0 |      0 |
|   20    |     43 |      0 |      0 |      0 |
|   21    |     50 |      0 |      0 |      0 |
|   22    |    142 |      0 |      0 |      0 |
|   23    |    153 |      0 |      0 |      0 |
|   24    |    168 |      0 |      0 |      0 |
----------------------------------------------------------------------
|                    Optimization results                            |
----------------------------------------------------------------------
|              Arc-Flow Standard - High Consumption                  |
----------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   24   |    KMeans |       Unlimited|  24447 | 21453 |     12.2%     |
----------------------------------------------------------------------
|                      Sailing costs                                 |
----------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
----------------------------------------------------------------------
|   24447 |   5511 |   6990 |      0 |   1446 |     4500   |    6000  |
----------------------------------------------------------------------
|                    Quantity discharged                             |
----------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
----------------------------------------------------------------------
|    1    |  -5560 |  -3000 |      0 |  -1500 |    -10060   |    No    |
|    2    |      0 |    600 |      0 |      0 |     600    |    No    |
|    3    |      0 |      0 |      0 |      0 |       0    |    No    |
|    4    |      0 |      0 |      0 |      0 |       0    |    Yes   |
|    5    |      0 |     10 |      0 |      0 |      10    |    Yes   |
|    6    |      0 |    890 |      0 |      0 |     890    |    No    |
|    7    |      0 |   1500 |      0 |      0 |    1500    |    No    |
|    8    |      0 |      0 |   1200 |      0 |    1200    |    No    |
|    9    |   1200 |      0 |      0 |      0 |    1200    |    No    |
|   10    |    900 |      0 |      0 |      0 |     900    |    No    |
|   11    |    750 |      0 |      0 |      0 |     750    |    No    |
|   12    |    150 |      0 |      0 |      0 |     150    |    No    |
|   13    |   2700 |      0 |      0 |      0 |    2700    |    No    |
|   14    |      0 |      0 |      0 |    750 |     750    |    No    |
|   15    |      0 |      0 |      0 |    450 |     450    |    No    |
|   16    |      0 |    450 |      0 |      0 |     450    |    No    |
|   17    |      0 |   1050 |      0 |      0 |    1050    |    No    |
|   18    |      0 |      0 |      0 |   1500 |    1500    |    No    |
|   19    |      0 |    450 |      0 |      0 |     450    |    No    |
|   20    |      0 |   1050 |      0 |      0 |    1050    |    No    |
|   21    |    600 |      0 |      0 |      0 |     600    |    No    |
|   22    |      0 |      0 |      0 |      0 |       0    |    Yes   |
|   23    |   2250 |      0 |      0 |      0 |    2250    |    No    |
|   24    |      0 |      0 |      0 |      0 |       0    |    Yes   |
Ship total|   8550 |   6000 |   1200 |   2700 |   18450    |          |
----------------------------------------------------------------------
|                      Time of visit                                 |
----------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
----------------------------------------------------------------------
|    1    |      0 |     90 |    101 |    112 |
|    2    |     31 |      0 |      0 |      0 |
|    3    |      0 |      0 |      0 |      0 |
|    4    |      0 |      0 |      0 |      0 |
|    5    |     58 |      0 |      0 |      0 |
|    6    |     82 |      0 |      0 |      0 |
|    7    |     90 |      0 |      0 |      0 |
|    8    |     29 |      0 |      0 |      0 |
|    9    |     44 |      0 |      0 |      0 |
|   10    |     80 |      0 |      0 |      0 |
|   11    |     36 |      0 |      0 |      0 |
|   12    |     33 |      0 |      0 |      0 |
|   13    |    104 |      0 |      0 |      0 |
|   14    |     30 |      0 |      0 |      0 |
|   15    |     40 |      0 |      0 |      0 |
|   16    |    132 |      0 |      0 |      0 |
|   17    |    138 |      0 |      0 |      0 |
|   18    |    125 |      0 |      0 |      0 |
|   19    |    163 |      0 |      0 |      0 |
|   20    |    168 |      0 |      0 |      0 |
|   21    |    158 |      0 |      0 |      0 |
|   22    |      0 |      0 |      0 |      0 |
|   23    |    168 |      0 |      0 |      0 |
|   24    |      0 |      0 |      0 |      0 |
```

Robustness

```
------------------------------------------------------------------------
|                    Optimization results                              |
------------------------------------------------------------------------
|            Arc-Flow Standard - 15% Minimum Inventory                 |
------------------------------------------------------------------------
Harbours| Clustering|   Compatibility|Solution| Bound |Integrality Gap|
|   19  |   Natural |       Unlimited|  13922 | 10098 |     27.5%      |
------------------------------------------------------------------------
|                      Sailing costs                                   |
------------------------------------------------------------------------
|   Total | Ship 1 | Ship 2 | Ship 3 | Ship 4 |  Ext. Feed | Ext. Sail|
------------------------------------------------------------------------
|   13922 |   6351 |   4265 |   1949 |   1357 |        0   |       0  |
------------------------------------------------------------------------
|                    Quantity discharged                               |
------------------------------------------------------------------------
| Cluster | Ship 1 | Ship 2 | Ship 3 | Ship 4 | Aggregated | External |
------------------------------------------------------------------------
|    1    |  -2300 |  -2200 |  -1300 |  -1500 |    -7300   |    No    |
|    2    |    900 |      0 |      0 |      0 |      900   |    No    |
|    3    |    500 |      0 |      0 |      0 |      500   |    No    |
|    4    |    400 |      0 |      0 |      0 |      400   |    No    |
|    5    |      0 |      0 |      0 |    500 |      500   |    No    |
|    6    |      0 |      0 |      0 |   1000 |     1000   |    No    |
|    7    |    800 |      0 |      0 |      0 |      800   |    No    |
|    8    |      0 |      0 |      0 |    300 |      300   |    No    |
|    9    |      0 |      0 |   1300 |      0 |     1300   |    No    |
|   10    |      0 |      0 |      0 |    400 |      400   |    No    |
|   11    |      0 |   2300 |      0 |      0 |     2300   |    No    |
|   12    |    600 |      0 |      0 |      0 |      600   |    No    |
|   13    |    400 |      0 |      0 |      0 |      400   |    No    |
|   14    |   1300 |      0 |      0 |      0 |     1300   |    No    |
|   15    |      0 |    900 |      0 |      0 |      900   |    No    |
|   16    |      0 |    400 |      0 |      0 |      400   |    No    |
|   17    |      0 |    800 |      0 |      0 |      800   |    No    |
|   18    |      0 |      0 |   1500 |      0 |     1500   |    No    |
|   19    |      0 |    800 |      0 |      0 |      800   |    No    |
Ship total|   4900 |   5200 |   2800 |   2200 |    15100   |          |
------------------------------------------------------------------------
|                      Time of visit                                   |
------------------------------------------------------------------------
| Cluster | Visit 1| Visit 2| Visit 3| Visit 4|
------------------------------------------------------------------------
|    1    |     58 |     67 |     76 |     81 |
|    2    |      0 |      0 |      0 |      0 |
|    3    |     32 |      0 |      0 |      0 |
|    4    |     38 |      0 |      0 |      0 |
|    5    |    126 |      0 |      0 |      0 |
|    6    |    113 |      0 |      0 |      0 |
|    7    |     50 |      0 |      0 |      0 |
|    8    |      0 |      0 |      0 |      0 |
|    9    |     56 |      0 |      0 |      0 |
|   10    |      9 |      0 |      0 |      0 |
|   11    |     70 |      0 |      0 |      0 |
|   12    |     83 |      0 |      0 |      0 |
|   13    |     91 |      0 |      0 |      0 |
|   14    |     97 |      0 |      0 |      0 |
|   15    |     32 |      0 |      0 |      0 |
|   16    |     25 |      0 |      0 |      0 |
|   17    |     14 |      0 |      0 |      0 |
|   18    |    126 |      0 |      0 |      0 |
|   19    |      0 |      0 |      0 |      0 |
```

Roald Hartvigsen

NTNU
Norwegian University of
Science and Technology