Including greenhouse gas emissions in a shipping company's decision making for the logistics of spare parts

A case study for chemical tankers

B. (Bas) Rossewij

Delft University of Technology

TU Delft

Stolt Tankers

Thesis for the degree of MSc in Marine Technology in the specialisation of Maritime Operations and Management

Including greenhouse gas emissions in a shipping company's decision making for the logistics of spare parts

by

B. (Bas) Rossewij

Performed at:

Stolt Tankers B.V.

This thesis MT.23/24.052.M is classified as confidential in accordance with the general conditions for projects performed by the TU Delft

To be publicly defended on November 24, 2023 at 10:00

Company supervisors	
Responsible supervisor:	Ir. W.H. Bodewes
Thesis exam committee:	
Chair/ Responsible professor:	Dr. ir. J.F.J. Pruyn
Staff member:	Dr. B. Atasoy
Company member:	Ir. W.H. Bodewes
Author details:	
Student number:	4547985
Institution: Project	Delft University of Technology
duration:	September, 2022 – November, 2023

An electronic version of this thesis is available at https://repository.tudelft.nl/.

Cover Image: Stolt Satsuki, (Stolt Tankers, 2022)

Preface

This report is written as the last step in obtaining my Master of Science degree in Marine Technology. I started this study out of a passion for combining technology and ships. This passion has grown into something that keeps me busy on a daily basis instead of only during the weekends. Throughout the years, I learned a lot about everything the shipping sector entails, especially during my time as a working student at Stolt Tankers. With the knowledge I received from my studies and seeing how this went in real life, I saw (and still see) a potential in improving the maritime industry by analysing what we are doing right now. By using the data about current processes, we all can improve the efficiency of our current processes. From a business perspective, money can be saved, but within the sector of transport, often this also reduces the carbon emissions.

This is why I am grateful that I was able to continue my professional journey at Stolt Tankers and that they provided me with an assignment to which I could put this line of thought into practice. Even before starting this project, I already saw many unnecessary transport movements for spare parts within Stolt Tankers. Within this thesis, I was able to analyse the impact on Greenhouse Gas emissions if these movements were optimised. Again, using everything that is currently available to improve from a business perspective and subsequently improve on Greenhouse gas emissions. Thanks to everyone at Stolt who guided me in finding the right thesis subject, especially Robert Zwart, for his time and effort. Also, thank you to everyone within Stolt who helped me get to the bottom of this problem, I added much value to my thesis.

During my time as a graduate intern in the Reliability and Performance team, I learned a lot. Not only on the topic of my graduation but especially about the processes onboard the ships. All this theory has even become a reality multiple times during ship visits, which really showed how the theory learned in university is applied in real life. Next to working hard, there was always room for (some) fun, which made me feel really welcome in the team. Thank you to all team members for the great time!

A special thank you goes to Wijnand Bodewes; it was a pleasure to be guided by him. Something that remarks Wijnand, is that he could really bridge the gap between the TU Delft and Stolt Tankers. His easy-going guidance throughout my thesis made that I was able to lead my project in the direction I preferred it to go whilst keeping an eye on the goals. Wijnand sometimes reminded me to take it easy after a deadline: recharge and then continue again. Wijnand ensured that even though researching is not my hobby, I still enjoyed this graduation project.

Besides the guidance at Stolt, I also received very valuable input on this thesis from Jeroen Pruyn, for which I want to thank him. Jeroen's patience throughout the process has really led to the final product. Sometimes, I was so focused on solving the problem that I lost track of how to write/implement the progress in an academic way. Jeroen, thank you for all your speedy replies to the emails and your academic guidance throughout.

Furthermore, I want to thank the people who had to endure the most during my thesis but were always there to support me. First, my friends, thank you for all the coffee breaks, which were sometimes a moment of relaxation but could also be a moment of realisation. In both ways, it has solved a lot of issues during the thesis. Then, my parents and sister, I know it has sometimes been hard to understand what was happening in Delft, but you have always supported me, and I know you are all very proud. Last but definitely not least, Anne-May. Thank you for always being patient and trying to listen in times I had a lot of stress. Spending time together always ensured I was more relaxed about graduation, which meant more to me than you can imagine.

B. (Bas) Rossewij
Rotterdam, November 2023

Summary

Shipping companies aim to reach the climate goals by following the rules from the International Maritime Oranization (IMO), which includes reducing Scope 3 emissions. Maintenance is a great contributor to the Scope 3 emissions of a shipping company, especially the transport of spare parts. The current state-of-the-art supply chain optimisation for the logistics of spare parts is based on costs and risks. This study aims to find the potential influence of including Greenhouse Gas (GHG) emissions in the decision-making process for the procurement of spare parts.

The element of the supply chain that makes it possible to plan when a spare part is needed is the maintenance policy. Planning in advance when a part is needed is possible for Preventive Maintenance (PM), which is therefore used throughout this research. However, integrating spare part management, which follows from this PM, with supply chain management has not yet been researched much. Similar findings have been made even when analysing spare part management in other industries; spare part management is not integrated much yet within the supply chain. This calls for an alternative approach to look at the spare part demand approach, as normally, this is addressed from the supplier's point of view. Analysing it from the shipping company's point of view means less data is available, especially when looking at a single spare part level. Therefore, this research uses the available data within a shipping company's Planned Maintenance System (PMS). The statistical first possible job date is determined by performing a statistical analysis on how much time a job date deviates from the PMS window.

This approach allows for the model to disregard the risks in the optimisation. This makes it possible to create a model that optimises between the freight cost, the cost of GHG emissions and the cost of capital. The model calculates each alternative's total costs and emissions using a brute-force approach. Alternatives consist of different delivery locations and different delivery dates. After calculating the total costs for each alternative, the model picks the most cost-efficient option.

The model is applied to a case study of a chemical tanker from Stolt Tankers B.V., using available data on its job history and historical location data. From the case study follows that including GHG emissions in the decision-making adds an additional 0.07% to 4.8% of savings in GHG emissions depending on the delivery strategy. When the delivery strategy is changed to delivery before the job, an improvement in GHG emissions can be achieved of 19.77% (at $0 per kg GHG). However, at the current price for GHG emissions, changing the delivery strategy reduces the possible savings in the total costs. When considering the change in delivery strategy, first, the strategy will be changed to delivery before with stock to analyse its performance. This has a significant cost impact with respect to the current delivery strategy. This is still the trade-off that should be made by the decision maker.

Contents

List of Figures

List of Tables

List of Acronyms

1 Introduction

Up to now, the Paris Agreement is the most significant agreement on climate change. ”Its goal is to limit global warming to well below 2°C, preferably 1.5°C, compared to pre-industrial levels” (United Nations Framework Convention on Climate, n.d.). Governments must reduce Greenhouse Gas (GHG) emissions to reach this goal. The shipping sector, however, is an international sector, meaning it is necessary to have only one party that determines the rules and regulations instead of separate national regulations. The separate nations are allowed to be more strict about these regulations. Therefore, the International Maritime Organization (IMO) strives to reach the goals set in the Paris Agreement by making rules that support this. The goals set by the IMO (International Maritime Organisation, n.d.-a) are clear, compared to 2008:

1. Reduce CO_2 emissions by at least 40% per vessel by 2030;
2. Pursue efforts to reduce CO_2 emissions by 70% per vessel by 2050;
3. Reduce the total annual GHG emissions of the complete fleet by 50% in 2050

Greenhouse gases emitted by companies or individuals can be divided into three categories (Ranganathan et al., 2004), also called scopes. A schematic overview of the different scopes is given in Figure 1.1. Scope 1 consists of all direct GHG emissions from sources owned or controlled by a company. Scope 2 includes the indirect GHG emissions from the generation of purchased electricity or heating energy. Scope 3 consists of all indirect GHG emissions other than those covered in scope 2. Scope 3 emissions are a consequence of the companies’ activity, but the company cannot directly control them.

According to Hertwich and Wood (2018) and Schmidt et al. (2022), on average more than 50% of the emissions from companies come from their supply chain (scope 3). Therefore, making it the most relevant part of the reduction of GHG emissions. Even though in the economic sector of transportation, the scope three emissions are ’only’ 45%, the impact that can be made by reducing scope three emissions is significant. Many companies in the transportation sector are now primarily focusing on direct emissions, which is most significant for them. However, looking at the indirect emissions could make a fair additional impact on reducing greenhouse gases.

To assess the environmental hotspots of a ship during its operational lifetime, Kjær et al. (2015) performed a Life Cycle Costing (LCC) based on a Life Cycle Assesment (LCA) using an Environmental Input-Output Model (EIO). The most significant contributor to GHG emissions is fuel combustion (scope 1), which takes 88.9% of all GHG emissions, which can also be seen in Figure 1.2. Even though the other categories (scopes 2 and 3) take on the rest of the 11.1% of the emissions, an impact can still be made by reducing these emissions. Currently, the IMO and the EU mainly target fuel and port emissions. However, most port emissions consist of the emissions necessary to enter a port or pass through a canal and thus cannot be changed significantly under the influence of a shipping company. Likewise, ”Other operational expenses” consist of many components relevant to the vessel’s crew, such as provisions and crew transportation. A category not usually addressed by the IMO or EU, but that has one of the highest $kgCO_2$/$ (see Figure 1.3) is maintenance. This category does not entail more than one

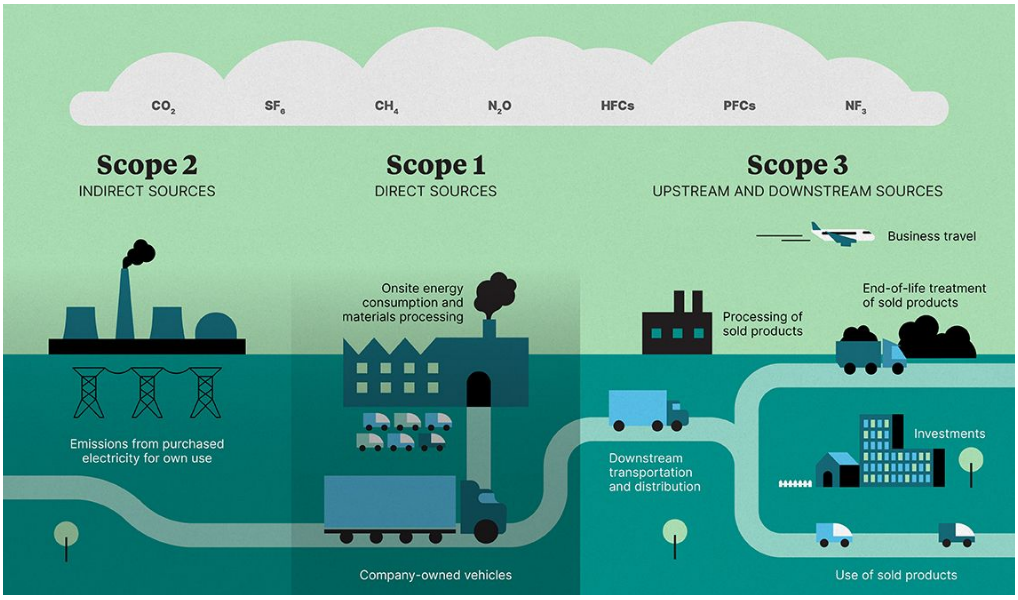


Figure 1.1: Classification of a company's emissions into three scopes (Oliver Wyman Forum, n.d.)

activity that results in the emissions and thus will be considered the most influential component in the emission of CO_2 (other than fuel) during the operational lifetime of a ship.

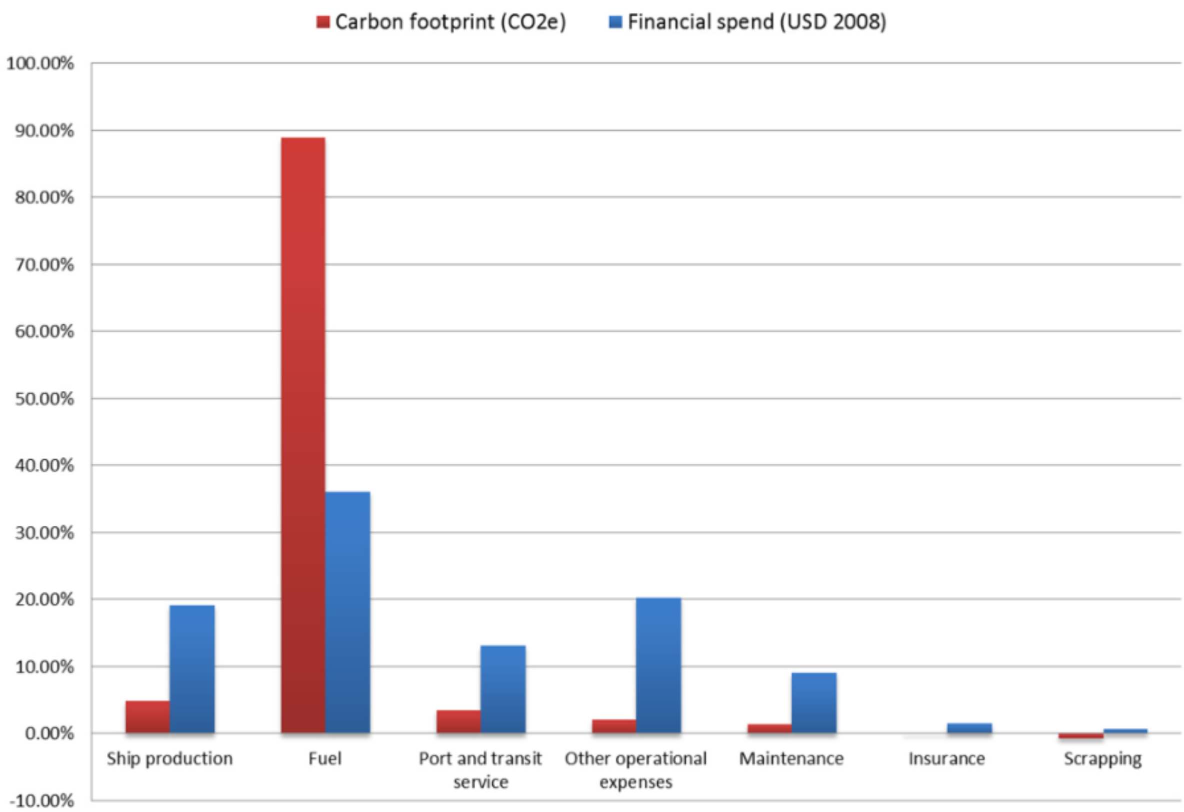


Figure 1.2: Relative share of CO_2 emissions and costs (Kjaer et al., 2015)

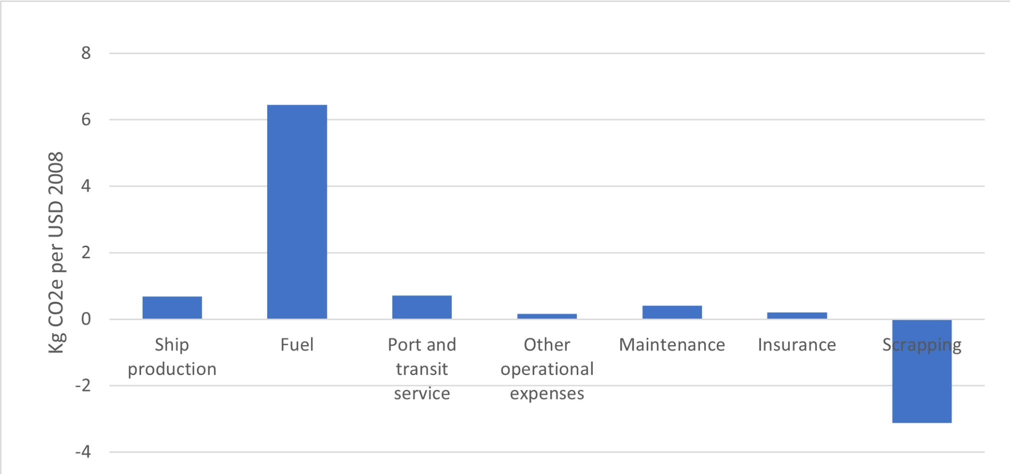


Figure 1.3: Kg CO_2-equivalent per USD for each category, based on Kjær et al. (2015)

Stolt Tankers B.V. (Stolt), one of the leading companies in chemical tankers and parcel tankers, owns 164 ships with 70 deep-sea ships (Stolt Nielsen, n.d.-a). By operating this fleet, Stolt has to keep in mind all the aspects of the company. One is to ensure that all ships are populated with a crew and that the machinery keeps working. The latter is achieved by regularly maintaining all systems on board for which spare parts are needed.
Stolt recognises the potential savings that can be achieved by looking at the indirect emissions from the transportation of spare parts. As spare parts are critical to the reliability of a ship's operation (Mouschoutzi & Ponis, 2022), many spare parts are transported worldwide to the always-moving ships. To reach their goal to reduce emissions by 50% in 2030 and be completely carbon-neutral in 2050 (Stolt Nielsen, n.d.-b), Stolt Tankers is also keen on improving on emissions that are made indirectly (scope 3). This focus on scope 3 is also stimulated by governments and the EU by using carbon taxes to stimulate companies to reduce their GHG emissions (European Commission, n.d.-a). Whilst this is not implemented specifically for scope 3 emissions, the prices for scope 3 activities are likely to increase with this taxation.
This, together with the fact that maintenance covers a significant part of the total operating expenses of a shipping company (Stopford, 2008), creates a good incentive for a company to look at all possibilities to reduce carbon emissions, even when they are indirect emissions. For shipping companies, the current procurement strategy is mainly cost- and risk-based. However, focusing on GHG emissions will also become more important for shipping companies, for environmental reasons, but also for costs.

1.1. Research goal

This thesis aims to find a way to include the GHG emissions in a shipping company's strategy and determine the influence of doing so by using the following research question:

"What is the potential reduction of Greenhouse Gas emissions that is yearly achievable by including indirect emissions into a shipping company's strategy for the procurement of spare parts?"

To guide the process of answering this question, the following sub-questions are defined:

1. What does a shipping company's current state-of-the-art supply chain optimisation entail?
2. Which modelling approach can be adapted to include Greenhouse Gas emissions in the procurement decision?
3. When are the spare parts required on-board the vessel based on historical- and preventive maintenance data?
4. How can the chosen modelling approach be applied to sustainable supply chain optimisation in the maritime industry?

5. What is the influence of the decision parameters on the final decision of the model?
6. What is the difference between the Greenhouse Gas emissions that have been emitted in the case year and the emissions as analysed by the model in the same year?

1.2. Report structure

The report is structured according to these sub-questions. Chapter 2 will elaborate on the current state-of-the-art supply chain within the maritime industry and the problem that is found here. This is combined with the available data from within the company, which highlights this problem. Next, in Chapter 3, it is explained which modelling approach can be adapted to include the GHG emissions in the procurement decision to make this sustainable. This also involves looking at when the spare parts are needed on board the vessel. Furthermore, Chapter 3 introduces aspects important to the decision-making process. These aspects are included in the chosen modelling approach, elaborated in Chapter 4, where the final model is explained and verified. To find the answer to the last two subquestions, the model is applied in a case study described in Chapter 5. The report will be critically reviewed in Chapter 6. Lastly, the answers to the sub-questions and the main research question will be presented in Chapter 7, together with the concluding remarks and recommendations.

2
Supply chain of the maritime industry

In this chapter, the supply chain of the maritime industry will be elaborated on, explaining the elements that influence the decision-making process. Hereafter, the available data within the case company, Stolt Tankers, is presented, as this presents parts of the problem. This chapter will also shed light on the different maintenance policies and how these influence the final decision.

2.1. Elements of the maritime supply chain

Vessels owned by maritime companies usually operate away from their home base and are continuously on the move. The availability of spare parts is essential to ensure the availability of the vessel (Eruguz et al., 2017a; 2017c). The fact that the voyage is often only planned shortly before the ship leaves a port complicates the procurement and ordering of spare parts. Mouschoutzi and Ponis (2022) adds that ”efficiently handling the supply chain and logistics of spare parts, from sourcing of the requested items to their delivery on board the vessel, is a major component of a successful maintenance strategy”. This section will elaborate on the different elements of the maritime supply chain.

To understand a logistical system of a company or a part of it, Huiskonen (2001) made an overview (given in Figure 2.1) to show the constituting elements of a logistics system. The different parts are described as follows:

Strategy/policies/processes: (from a customer’s point of view)	Make sure spare parts are available and of proper quality. This includes comparing different suppliers and deciding on the supply strategy.
The network structure:	Defines the number of inventory echelons and locations used in the system.
Management of relationships:	Managing cooperation, distribution of responsibility and risk sharing between the involved parties.
Coordination and control:	Decisions about inventory control, performance measuring (including incentive systems), and information systems to implement control procedures.

Huiskonen (2001) explained the most relevant control characteristics, which are displayed in Figure 2.2. Combined, these four characteristics also define the inventory management system (Mouschoutzi & Ponis, 2022). All four control characteristics have their influence on the elements of the logistics system elements. Therefore, each spare part’s logistics system will differ based on its characteristics.
The first is *criticality*, which is related to the consequences of when the spare part is not readily available. A practical way to determine the criticality can be expressing it in time in which the failure of a component needs to be corrected. This is important not only for a supplier but also for the inventory management system because ship owners would prefer to have a critical component in their inventory on board (Mouschoutzi & Ponis, 2022).
A control characteristic specific to maintenance spare parts is *specificity*, which relates to the difference

between standard- and equipment-specific parts. Most often, equipment-specific parts are more challenging to acquire than standardised parts because the manufacturers often stock standardised parts. Another relevant control characteristic is *value of a part*. The higher the price of a specific part, the less interesting it will become to keep it in stock. Next, administrative costs of cheaper parts will be relatively high compared to the item price. Therefore, buying multiple of the same parts will be more cost-efficient.
The availability of stock from the supplier will also increase when there is a clear *demand pattern*, especially when there is a high demand for a large volume of parts. The demand for spare parts comes from the number of maintenance operations performed, which again is influenced by the maintenance policy (Mouschoutzi & Ponis, 2022). Section 2.4 elaborates on this.

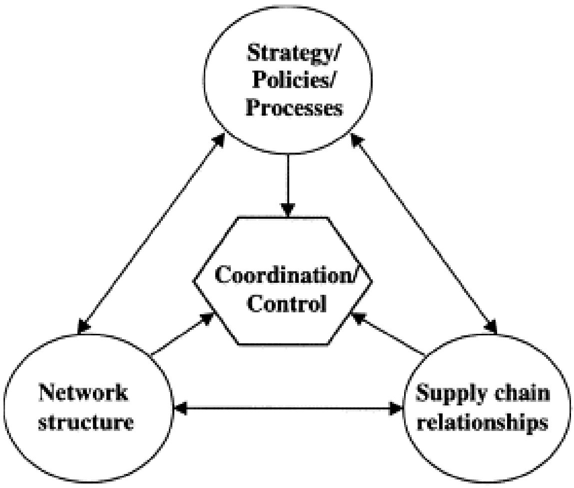


Figure 2.1: The constituting elements of the supply chain (Huiskonen, 2001)

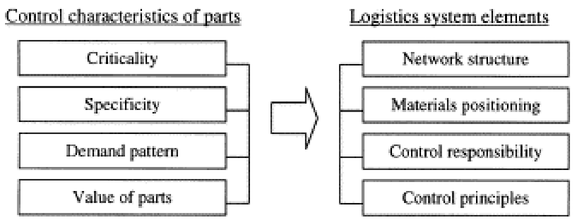


Figure 2.2: Relevant control characteristics and logistics systems (Huiskonen, 2001)

With this information, an overview is created of the regular supply chain elements of a shipping company. This information is needed to know which information should be used from within a company to tackle the problem as stated in Section 1.1. In summary, all control characteristics displayed in Figure 2.2 are needed as well as the maintenance strategies that are applied to the specific spare parts.

2.2. Available data

To effectively utilise the available data within Stolt Tankers, it is necessary to preprocess the data related to the procurement of spare parts. This involves combining multiple datasets to determine when spare parts have been used, which is performed using Python. The used datasets, as shown in Table 2.1, are as follows: "Material Consumption", which provides details about the quantity of each spare part used for specific jobs; "All Spare Parts", which serves as a reference list to ensure that all materials used in the combined dataset are categorised as spare parts, it also contains information about the parts; "Job History", which contains records of job execution along with unique job codes in the format "yyWO-xxxx" (yy representing the year number and "xxxx" representing the job number); "Global Purchase History", which contains information about the purchase of the material including delivery ports, dates and prices; "Counter Values", which contains historical information on the running hours of different equipment; and "Supplier Locations" which contains the locations of all suppliers, also expressed in coordinates. These datasets combined provide all available data as needed accroding to Section 2.1. How all datasets are connected is depicted in Figure 2.3 and explained throughout this section.

Connecting the "All Spare Parts" dataset and leaving out the datapoints from the "Material Consumption" list where no corresponding spare part is found ensures that materials that do not classify as spare parts are excluded from the dataset. An example of such parts can be consumables intended to be used during regular operations and therefore replenished more regularly.
As said, the "Material Consumption" dataset contains information about the number of spare parts that

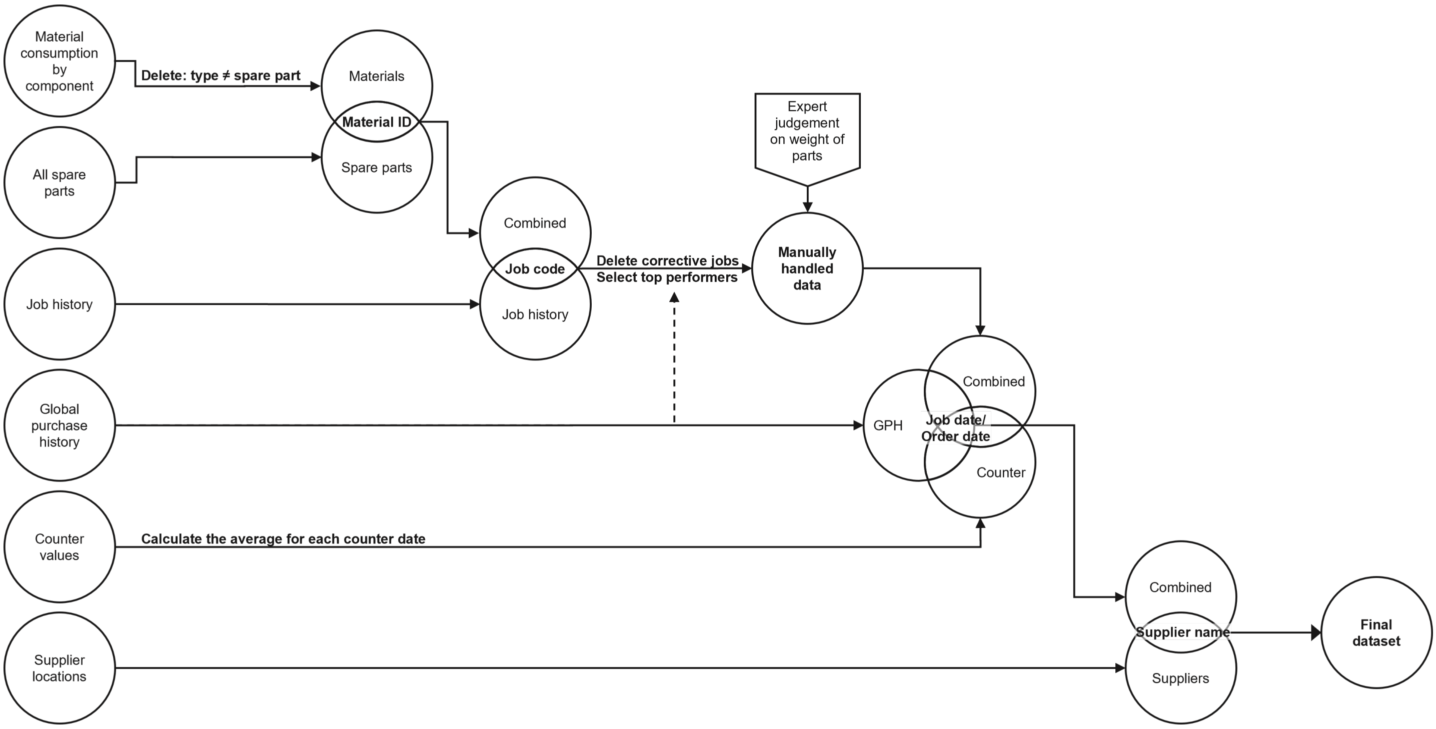


Figure 2.3: Overview of how the available data is utilized to form the final dataset (full size in Appendix B)

are used for a specific job. However, when going through this dataset, a significant amount of zero-quantities have been found: 72%. After conferring with people working with the software from which the data is retrieved, it appears that this is mainly a reporting issue. Hence, for most cases, assuming that at least one material has been used for the job when it is indeed linked to it will be valid. Therefore, this is implemented in the dataset ”Material Consumption”. Later is explained how this is verified using the ”Global Purchase History” dataset.

The ”Job History” dataset contains, as said, all records of performed maintenance jobs, including those related to corrective maintenance. Consequently, the entries associated with corrective maintenance jobs must be removed from the dataset.

The data reporting issue prompts the need to analyse the data quality before picking a ship to analyse for the case. Therefore, the combination of the three datasets is conducted on a randomly selected group of 33 from the total 103 vessels within Stolt Tankers. The reason for not analysing all ships is due to the amount of manual labour required to export all data. Next to that, in the end, only one ship will be used as a case for this report. The randomly picked ships come from different ship classes and trading areas. Upon comparing the final usage list with the ”Global Purchase History” dataset (see Table 2.1), a disparity is indeed observed between the number of ordered spare parts and the number used. A similar pattern emerges regarding the count of unique spare parts used and the requested unique spare parts. This disparity is mostly caused by the difference in reporting behaviour onboard the ships. The top 5 performers regarding total usage compared to the total number ordered are displayed in Table 2.2, and a complete overview is given in Appendix A. In all overviews, the real ship names have been replaced with a vessel number that is known within Stolt Tankers. Based on this comparison, T0126 has been picked to be analysed throughout this report because it has the most available data compared to what is actually ordered. T0126 presented a usage of 86.51% of the ordered parts during its lifetime. The dataset is then consolidated based on unique spare part names. To ensure an adequate sample size for each spare part, spare parts with fewer than five data points were excluded from this research.

To be able to say something about a ship operating worldwide and receiving spare parts from suppliers located globally, the ship’s operational profile is analysed. In Figure 2.4, the operational profile for T0126 is depicted. As can be seen, the ship has traded worldwide over the past years (for which the data is available). When the operational pattern is analysed for each individual year during this time

Table 2.1: Available datasets on the procurement of spare parts within Stolt Tankers (connection shown in Figure 2.3)

Dataset	**Explanation**
Material Consumption	*Contains information about which and how much spare parts are used for which job*
All Spare Parts	*Contains all spare parts present onboard the vessel, including information about the maker/supplier or the criticality*
Job History	*Contains information about all jobs performed on the vessel over the years. This dataset also contains information about the preventive maintenance times and when the jobs have been executed*
Global Purchase History	*Contains all information about the spare parts that have been bought for a vessel and where and when it is delivered*
Counter Values	*Contains the historical counter values for all equipment*
Supplier Locations	*Contains the locations of all suppliers within Stolt Tankers*

Table 2.2: Top five vessels with regards to the reporting quality, full table in Appendix A

Vessel Number	**Percentage of total ordered parts used**	**Percentage of total unique ordered parts used**
T0431	*87.18%*	*39.84%*
T0126	*86.51%*	*55.02%*
T0127	*65.72%*	*38.20%*
T0172	*60.27%*	*131.53%*
T0429	*60.03%*	*26.05%*

frame, there is one year in which the vessel only sailed between Europe and the United States.

Then, the delivery port is added to the corresponding spare parts from the ”Global Purchase History” dataset. Within Stolt Tankers, no information exists on when the used part was ordered or arrived onboard the vessel. So the connection between the ”Global Purchase History” and the ”Job History” has to be made on the basis of the closest delivery date to the job date. Usually, spare parts are used whereafter they will be ordered for restocking. This is also valid for critical spare parts, which, in practice, will be restocked as soon as possible after usage. Based on company experience, it is found that in specific cases, the spare parts are delivered prior to the due date for two main reasons. Firstly, this occurs when the spare parts are typically not stocked onboard. Secondly, some suppliers have prescribed ordering the parts before the due date (L. Teerling, personal communication, June 2023). Therefore, the exceptions made on a dataset differ depending on the equipment that is used onboard the vessel and on the data that is analysed.

In some cases, there is no purchase order before or after the date on which the job was performed. Then, the closest date (to the job) is taken as the delivery date with a corresponding delivery location. If there is no delivery date available at all, the spare part is not registered in the ”Global Purchase History” and thus has no delivery location. Hence, this part is then disregarded from the dataset.
The literature review explains that the spare parts chosen for the analysis should differ in supplier, supplier country, weight, criticality and replacement time. Information missing in the Stolt Tankers database is the weight of the spare parts and the location of the suppliers. The weight has to be added manually. The remaining dataset is consolidated based on the description of a spare part and the manufacturer, simplifying for easier navigation when adding the weight manually. This list is then appended with the weight of each item. As this information is unavailable within the Stolt Tankers databases, two experts are consulted to assess the weight category of the items. The experts consulted are Lennert Teerling (Reliability & Performance Manager at Stolt Tankers) and Evgeny Sviridov (Superintendent of, amongst others, T0126 at Stolt Tankers) (personal communication, May 2023). The interviews are held individually, so the two experts are not influenced by the other’s answers. For all remaining items in the dataset,

Figure 2.4: The operational profile for T0126 between April 2018 and July 2023

the experts provided the weight with a difference of more than 1kg for 30 items. A threshold of 1kg is used, as this is suspected to be influential compared to the deviation already coming from estimating the weights. The items with an estimation differing more than the threshold are either verified by letting the two experts discuss them or by using an external source. For example, where one expert mentions the weight for just a filter, the other says it is the whole part, including housing. When looking again, they agreed that this part is the filter only.

The counter values have to be added to this list of spare parts. This is done by looking at how a prediction was made within the Planned Maintenance System (PMS). The PMS estimates the next due date from a counter based on the average running hours over a period. The new average is calculated according to Equation (2.1) and can be converted to a predicted number of preventive maintenance days using Equation (2.2). This value is what the PMS used at that moment to forecast when the next job was due. By looking at the job date from the dataset, the predicted date (as in the PMS) is added to the final dataset by taking the previous date added with the result from Equation (2.2).

$$\begin{aligned} &New\ average\ value = \\ &\qquad \frac{New\ counter\ reading - Previous\ counter\ reading + 30.5 \cdot Previous\ average}{New\ reading\ date - Previous\ reading\ date + 30.5} \end{aligned} \tag{2.1}$$

$$PM\ days = AVG\ RH \cdot RH_{PM} \tag{2.2}$$

The last data set to add is ”Supplier Locations”, which contains the name of the supplier connected to their location. This location is expressed as an address but also as coordinates. Based on the supplier name, it can be connected to the supplier, adding the location to the dataset. This results in a dataset with varying characteristics, which are displayed in Section 2.2.1.

2.2.1. Dataset characteristics

This section displays the dataset’s characteristics for the data that is retrieved for the purpose of this study. As mentioned before, the dataset consists only of a subset of the total available data. The dataset has a variation in makers of the spare parts, meaning there are different supplier locations available in the dataset. In Figure 2.5, the variation in supplier locations is shown. It can be seen that a significant amount of the spare parts (~63%) are supplied from Rotterdam in the Netherlands. Next to that, for the other suppliers, the Rotterdam Port would also be the closest delivery option. Furthermore, Rotterdam is the main storage hub for Stolt Tankers. Therefore, it would be logical if most spare parts were delivered to the Port of Rotterdam. However, in Figure 2.6, it can be seen that also a significant amount of spare parts is delivered to the Port of Singapore (SGSIN), from which, according to Figure 2.5, almost no parts were supplied.

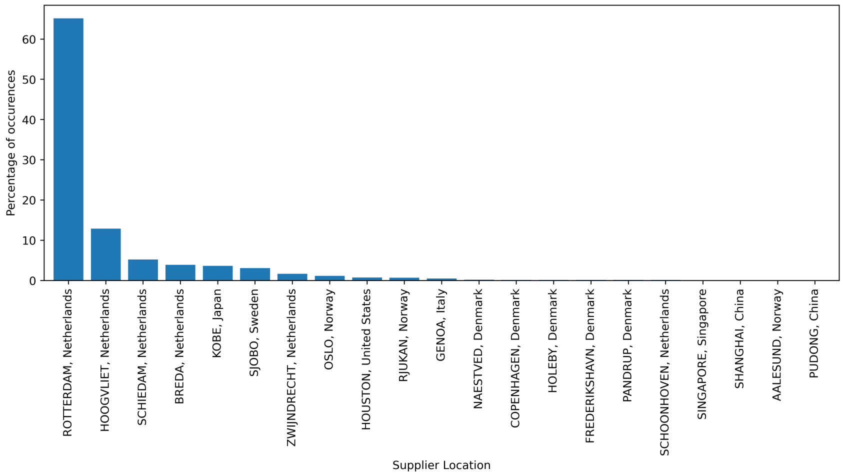


Figure 2.5: Occurances of the different supplier locations

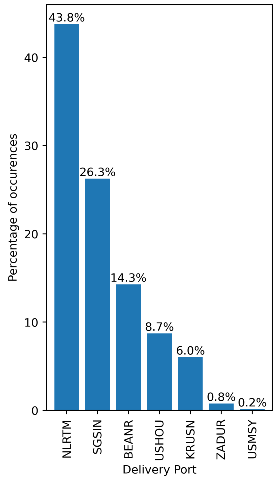


Figure 2.6: Occurences of spare parts being delivered at a port

According to Section 2.1, the criticality of the spare parts is one of the control characteristics of the supply chain. Therefore, this is included in the dataset. As seen in Figure 2.7, most spare parts are non-critical. Next to non-critical and critical spare parts, there are also redundancy-reliant spare parts. These spare parts become critical when the last one in stock is used, then the spare part should be delivered to the ship as soon as possible.

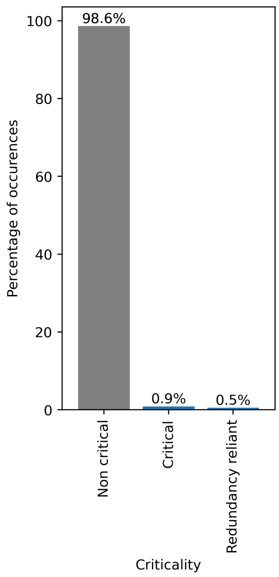


Figure 2.7: Occurrences of criticality classification in the dataset

The weight of the items in the dataset varies between 0.1kg and 25kg and consists of 19 different weights that can be analysed. In Figure 2.8, it is shown that items of 0.1kg have been delivered the most to the vessel.

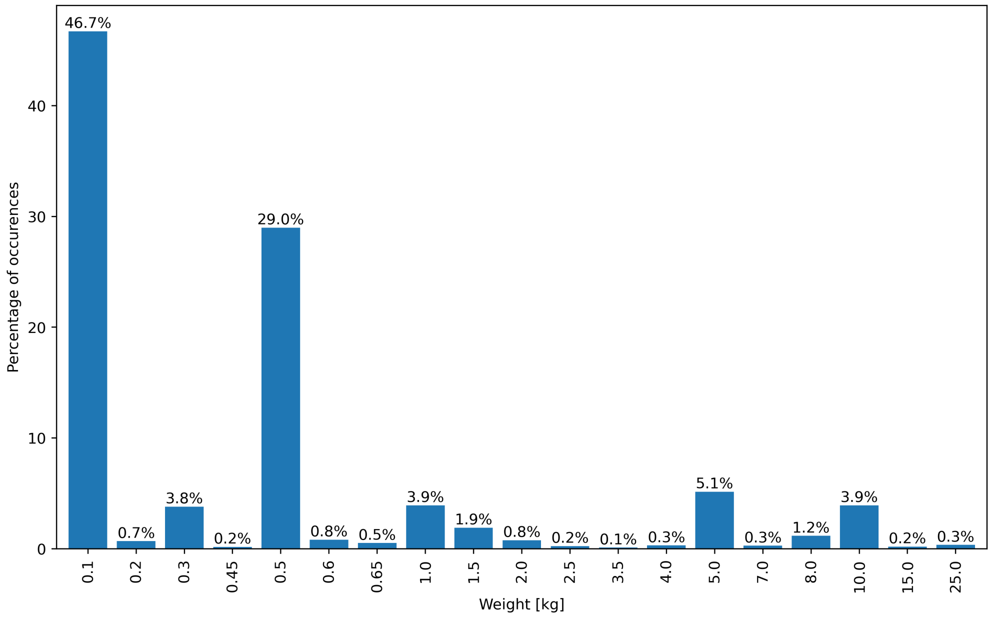


Figure 2.8: Occurrences of different weight within the dataset

The preventive maintenance jobs in the dataset are required for different components onboard the ship. They have intervals based on either running hours or on a fixed date. Within the dataset, there is also a variation present in the specified interval; there are 23 different intervals attached to the jobs. This can be seen in Figure 2.9 and the fact that there are time- and counter-based intervals.

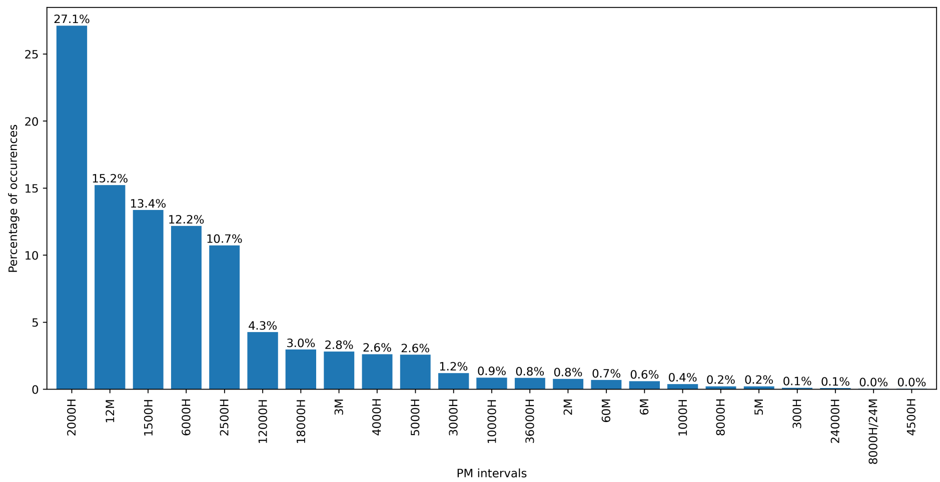


Figure 2.9: Occurences of the Preventive Maintenance intervals

As will be explained in Section 2.3, it is essential to know the demand pattern corresponding to the dataset. In Figure 2.10, an example of a demand pattern of a subset of the data is shown. Due to confidentiality, the other plots of the dataset are not shown in this report. What can be concluded from the demand pattern plots is that for most spare parts, there is only a small amount of data available, and the data is intermittent, meaning there are a lot of pauses between the occurrences. The length of these pauses also has a variance in them. When looking at all spare parts for this vessel, the demand pattern tends to be a little bit more in the direction of erratic demand. That also comes from the fact that there is a major storing moment approximately every three months, which happens in one of the main ports. This can also be recognised from the graph in Figure 2.11. Mind that this graph does not show the months without deliveries.

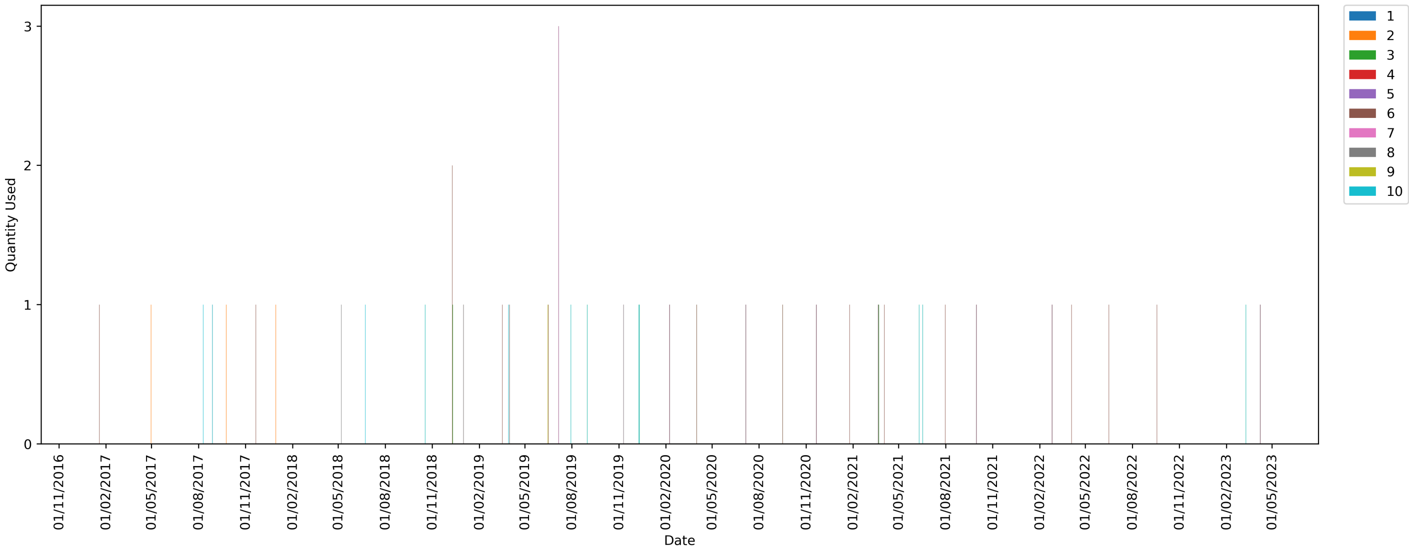


Figure 2.10: Demand pattern of a subset of the data of T0126, based on usage of spare parts

2.3. Spare part demand

In Section 2.1, it is explained that one of the control characteristics is the demand pattern, which follows from the usage of spare parts. As mentioned in the introduction, within Stolt, there are two ways of implementing preventive maintenance: based on supplier specification or historical data. From the aviation industry, it is found that historical data can be used to improve the preventive maintenance window as given by the supplier (Feng et al., 2021). To be able to perform the maintenance, the spare part should be available onboard the vessel. According to Ilgin and Tunali (2007), the influence of maintenance policies on the spare part provisioning policy cannot be ignored. The demand characteristics of spare parts are also what makes forecasting complicated. These characteristics are, according to Babaveisi et al. (2022):

1. Type of demand (visualised in Figure 2.12):
   - Intermittent (irregular demand with low demand quantity variations)
   - Lumpy (irregular demand with high demand quantity variations)
   - Erratic (regular demand with high demand quantity variations)
   - Smooth (regular demand with low demand quantity variations)
2. Dependence on descriptive factors (factors related to maintenance and repair, and working condition that affects the failure rate, i.e. the demand)

Feng et al. (2021) also mentions these four characteristics and adds that solving realistic questions with an increasing demand pattern is hard.

Literature on spare parts demand forecasting can be divided into three major categories: time-series forecasting, contextual forecasting and comparative studies (Pinçe et al., 2021), as can be seen in Figure 2.13. Mouschoutzi and Ponis (2022) also denotes the same two ways of demand forecasting methods but describes them as time-series and reliability-based forecasting. Time-series forecasting methods mainly rely on historical data and therefore do not need contextual information, such as expert judgement or product characteristics. From the comparative studies, Pinçe et al. (2021) concludes that there is no best forecasting algorithm. However, during the literature review, is found that historical data is useful for improving maintenance forecasting.
Within time-series forecasting, nonparametric approaches generally outperform the parametric methods (Pinçe et al., 2021). Nonparametric approaches consist of bootstrapping, empirical methods, and

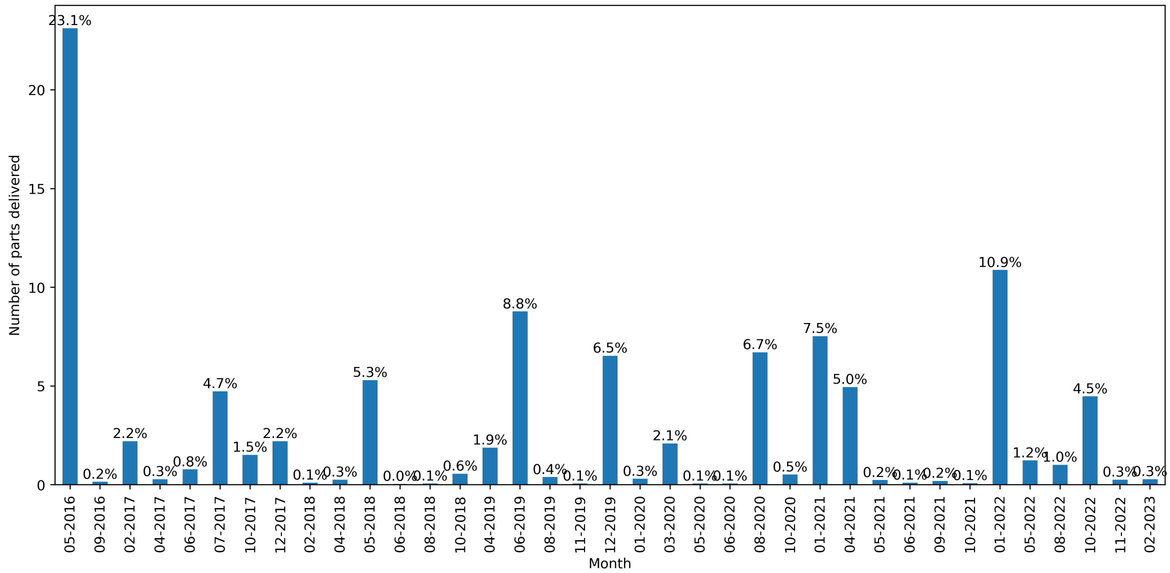


Figure 2.11: Number of parts delivered to the ship per month

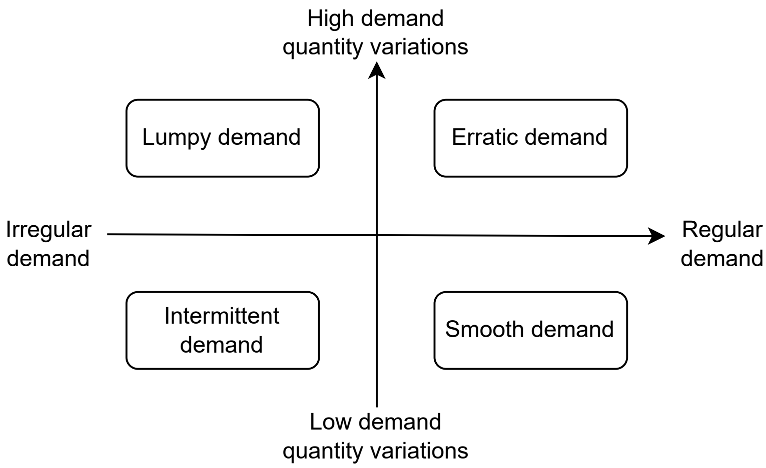


Figure 2.12: Different types of demand as described by Babaveisi et al. (2022) (own figure)

neural networks. A review written by Hasni et al. (2018) points out that nonparametric approaches are being applied more often, as a distribution-free approach would not lead to misleading information (when the wrong distribution would have been picked). Within the nonparametric approaches, bootstrapping has drawn a lot of attention in recent decades (Hasni et al., 2018).
According to Babaveisi et al. (2022): "Parametric approaches consider the demand over the lead-time as a predefined parameter with a known probability distribution (e.g. normal, Poisson), while non-parametric approaches extract the distribution from the data". Pinçe et al. (2021) also notes that a non-parametric method includes (historical) data to derive the demand distribution in contrast to parametric approaches. This makes the nonparametric approaches better suited to determine the future spare part demand, as Feng et al. (2021) suggested to include the historical data in the preventive maintenance window.

Mouschoutzi and Ponis (2022) states that parts for which known maintenance or inspection windows are termed x%-parts. This is because only x% of the preventive jobs will really require spare parts. For corrective maintenance, demand timing and quantities are unknown in advance, so then the demand is stochastic. It is also mentioned that using advanced demand information, such as a schedule of planned inspections, can increase demand forecasting accuracy. Also, the failure rates (λ), Mean Time

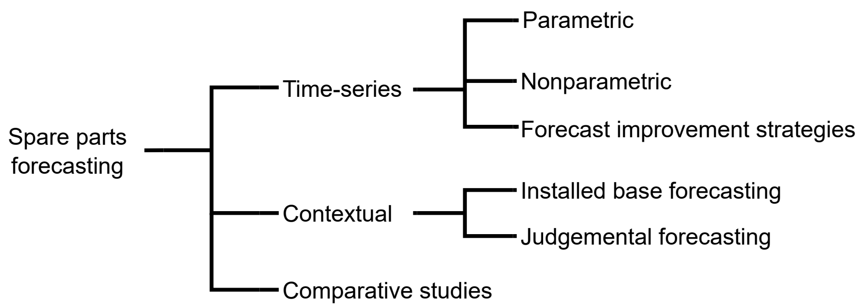


Figure 2.13: Literature on different spare parts forecasting approaches, adapted from Pinçe et al. (2021)

Between Failures (MTBF), and Probability of Failure (PoF), can assist in forming better maintenance planning and thus spare part planning. Often, the preventive jobs are scheduled such that a part is replaced before the MTBF. Mouschoutzi and Ponis (2022) notes that different spare parts have different underlying demand patterns and thus require other forecasting techniques. First, a demand classification should be performed to find the appropriate forecasting approach. Two factors that are most often used in demand classification are the Average Demand Interval (ADI) and the Coefficient of Variation (CV^2). Because there are many different spare parts to be analysed, these can not be looked at as a group but should be considered individually. However, for the methods described before, such as those from Feng et al. (2021) or the bootstrapping method, there is a need for enough data (more than 50 data points (Scikit learn, n.d.)). Based on the dataset characteristics found in Section 2.2.1, there is not enough data available to apply these methods in order to make a proper prediction. This asks for an alternative approach to determine when a spare part is needed.

2.4. Maintenance Policies

Before finding an alternative approach to determining the demand, another element of the supply chain that should be clear is the (maintenance) policy. This section displays maintenance policies to create an overview of the different policies. The applied maintenance policy influences the predictability of when a spare part is needed, next to the replacement time and classification of the spare part. Furthermore, the maintenance policy greatly affects the availability of the vessels owned by a shipping company (Tinga, 2013; Turan et al., 2009), hence knowing which maintenance policies there are is essential for the remainder of this report.

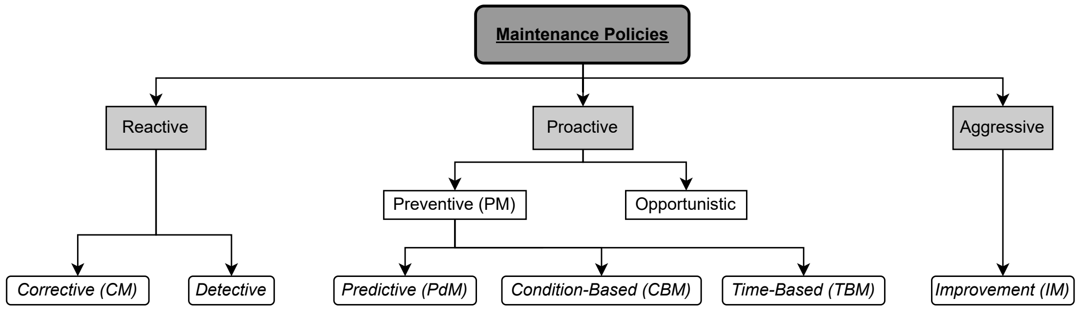


Figure 2.14: Overview of the maintenance policies, adapted from Tinga (2013)

In Figure 2.14, a schematic overview is given on how Tinga (2013) classifies the different maintenance policies. As can be seen, the maintenance policies can be divided into three categories: Reactive, Proactive and Aggressive. In other literature (Gandhare & Akarte, 2012; H. Wang, 2002; Yang, 2003), three types of maintenance are pointed out as common: Improvement Maintenance (IM), Corrective Maintenance (CM), and Preventive Maintenance (PM).

Aggressive maintenance or IM is performed to try to eliminate failures. This can already be done during the vessel's building or during its operational lifetime by replacing something with an improved version. Gandhare and Akarte (2012) and Tinga (2013) refer to a well-known example of IM as Total Productive Maintenance (TPM), which makes the method of IM more specific by continuous improvement of the system which leads to less maintenance.

Reactive maintenance is characterised as repairing or replacing a part after it is broken. This policy is mostly recognised as Corrective Maintenance (CM) (Cassady et al., 2001; Tinga, 2013). Detective maintenance only applies to hidden or unrevealed failures which come to light after a periodic test of this part. Cassady et al. (2001) describes CM as: "to perform a minimal repair on a failed component".

Proactive maintenance is performed before the element is expected to fail. In addition to CM Cassady et al. (2001) also gives two other maintenance policies that are available to the decision maker, which is: "to repair- or to replace a working component". This is often referred to as Preventive Maintenance (PM) in literature. PM intends to keep the equipment in good operating condition and changes a component when there is an indication that it is about to fail (Yang, 2003). If the owner does PM on its assets, possibly expensive maintenance can be prevented compared to CM (Turan et al., 2009). Gandhare and Akarte (2012), Tinga (2013), and Yang (2003) point out that two approaches are often used for PM, which are Time Based Maintenance (TBM) and Condition Based Maintenance (CBM). TBM is scheduled in advance based on either running hours or calendar time to prevent failure. If there is no failure in between, the component is replaced after a certain time. Ilgin and Tunali (2007) also mentions that this type of Preventive Maintenance (PM) is scheduled maintenance. Therefore, it will be relatively easy to determine when which parts will be needed. Contrary to TBM, CBM is a type of PM where the condition of a component is measured or inspected to see whether it needs replacement. This can be performed in real-time (using sensors) or periodically (physical inspection) after a set time. The latter should not be confused with TBM. According to Pahl (2022), these physical inspection techniques include, for example, vibration analysis, acoustic emissions, ultrasonic testing implementations, oil analysis, strain measurement, electrical effects, shock pulse method, radio-graphic inspection, and thermo-graphic monitoring technology.
Arena et al. (2022), Lazakis et al. (2010), and March and Scudder (2019) mention Predictive Maintenance (PdM) as a way to dynamically manage PM, by using real-time data analytics. With the increasing use of Big Data and upcoming technologies such as Internet of Things (IoT) supporting this, PdM is currently one of the most prominent approaches for data-driven monitoring of industrial systems to maximise reliability and efficiency. PdM implements both TBM and CBM to find the right timing to perform maintenance to a specific part.
The other component of proactive maintenance is Opportunistic maintenance. Here tasks on a specific (sub)system are triggered by other tasks performed in a neighbour (sub)system (Tinga, 2013). Even though the other subsystem might not require maintenance, this approach has some advantages, such as reduced transportation and labour costs.

Another way to look at maintenance is explained by Pahl (2022) and Stopford (2008), who divide maintenance actions into routine and periodic maintenance. This, however, is not a policy but a separation between the different kinds of maintenance work that can be carried out. Both routine maintenance and periodic maintenance can be regarded as planned maintenance.
Routine Maintenance consist of the routine maintenance which can be carried out while the ship is at sea. This includes tasks such as painting the superstructure or carrying out steel renewals. Routine maintenance also includes repairs after a breakdown of equipment and the replacement of parts by spare parts when needed. Routine maintenance is performed when the ship is still in use.
Periodic Maintenance covers the part of the planned maintenance performed when the ship needs to go into a dry dock, meaning the ship will temporarily be unavailable. This includes major machinery replacement or repairs and maintenance to the vessel's hull.

2.4.1. Influence of the maintenance policy

The influence of maintenance policies on the spare parts provisioning policy cannot be ignored according to Ilgin and Tunali (2007). The next part will show the impact of the different policies mentioned throughout this section.

When a repair-by-replacement strategy is used, the spare part must be on board the vessel to allow the replacement. The spare part should either be delivered to the vessel or already be on board (as mentioned in Section 2.4). It can be delivered using an aircraft, helicopter, boat, truck or other dedicated vehicles. This can be rather expensive. However, as explained in Section 2.4, holding it on board the vessel can also be costly based on several factors, including the spare part's value, the risk of obsolescence and the limited space available in the vessel's store (Eruguz et al., 2017a; 2017c). Spare parts critical to maintaining the ship's (safe) operation will always be on board.

When performing Condition Based Maintenance (CBM), which is increasingly common in the maritime industry (Mouschoutzi & Ponis, 2022), failure warnings are given when a part is expected to fail based on its condition and the monitored degradation rate (see Section 2.4). This makes it possible to optimise the supply chain by integrating the condition of parts into the supply chain decisions (Eruguz et al., 2017a; Kian et al., 2019; Zhao & Yang, 2018).

Preventive Maintenance (PM) could have a disadvantage because it presents a "lumpy demand" scenario, in which a significant amount of slow-moving units are required at a specific time. Even if preventive maintenance is performed, the needed amount of spare parts is stochastic. However, if previous inspections are performed, a better prediction of the required spare parts can be made compared to Corrective Maintenance (CM). In both cases (PM and CM), safety stock is needed onboard to prevent downtime, while relatively late orders cause this lumpy demand scenario (Vaughan, 2005; W. Wang & Syntetos, 2011). The reason for needing safety stock is that additional factors, such as human error, can cause CM to be required to keep the ship operational (Cullum et al., 2018). There will be a balance between PM and CM, which will consider both risks and costs for the company (Lampreia et al., 2022) However, when PM is assumed to be scheduled entirely, the demand for spare parts can be predicted (Ilgin & Tunali, 2007).

Yet there has not been much research on integrating spare part management into supply chain management, even though it is considered an essential part of the supply chain (Anglou et al., 2021; Pahl, 2022). The issue around spare parts is rather complex due to the ship's operational environment and the expected reliability and safety required onboard a ship.

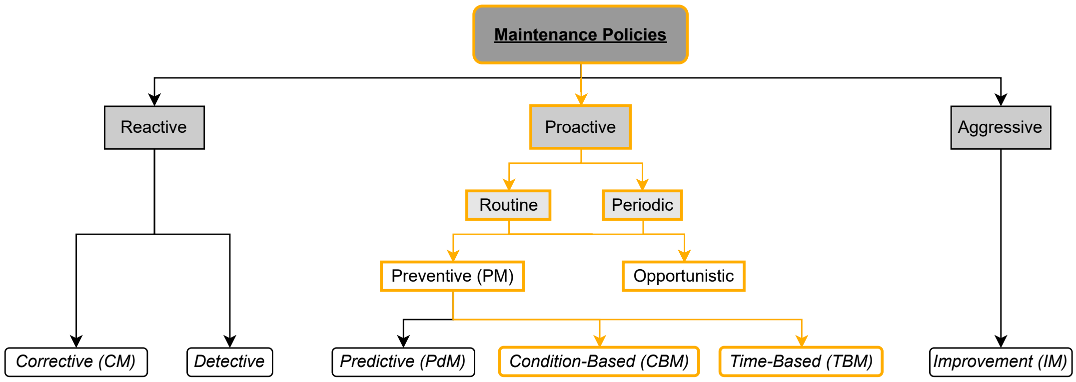


Figure 2.15: Extended overview of the maintenance policies as given in Figure 2.14, adapted Tinga (2013)

In conclusion, knowing how often and when to replace a particular part with a spare part depends on the applied maintenance policy. This also affects the predictability of when a spare part is needed. The applied delivery policy influences the total cost of replacement. On the one hand, keeping stock of a part costs money, but making a delivery away from a port also brings additional costs. This tradeoff should be made while considering the criticality of a spare part. To be able to know when a spare part is needed onboard the vessel, and based on the available data within the case company, this research focuses on the predictable Preventive Maintenance (PM). This is also highlighted in the overview of maintenance policies given in Figure 2.15.

2.5. Current state-of-the art

A more in-depth review has been performed to understand better the supply chain and how the requirements can be determined. It is also found that an integrated approach, where all elements of the supply chain are regarded, yields better results in the optimisation of the supply chain. Therefore, the supply chain elements are also explained in this chapter. One of the elements of the supply chain is the demand pattern, which is influenced by the applied maintenance policy. The optimal maintenance policy is gathered by considering cost and risk. Lastly, how and when the spare parts are bought, also called procurement, mainly depends on the applied maintenance policy. Hence, the maintenance strategy is important for this research.

To know when a spare part is needed onboard the vessel, maintenance planning is important. As explained in this chapter, the available data on the usage of spare parts on the user side of the supply chain is not sufficient to make a prediction. Therefore, the next chapter presents an adapted approach to the current supply chain, where the Planned Maintenance System (PMS) is used to predict the spare part usage. Still, the decision maker should make the trade-off between cost and risk. Risk is often considered a function of time and the cost associated with the vessel's downtime. According to Moussault and Pruyn (2020), including risk in maintenance optimisation requires a lot of operation data, which substantially complicates the matter. Therefore, the decision-making process for this thesis should be able to deliver the spare part on time such that the risk is (almost) eliminated. Almost, because it will be hard to eliminate the risk of a breakdown completely. Doing so requires the model to know when to deliver the spare part well in time. This is highly dependent on the maintenance policy and the vessel's location. The latter determines the transportation time to the vessel and, thus, the supply chain planning. This is all explained in the next chapter.

3

Sustainable decision making

Chapter 2 discusses the requirements to make a supply chain decision while taking into account the elements of the supply chain. A trade-off between different options will be made based on costs and risk. Next to the current criteria of the supply chain, as were discussed in Chapter 2, this study aims to include Greenhouse Gas (GHG) emissions into the decision-making process. To be able to determine a suitable modelling approach to include GHG emissions in the decision-making this model first elaborates on the decision-making process. Next, as required from Chapter 2, a different approach to determine the earliest delivery date is explained. After having picked a modelling approach, this chapter will introduce the remaining information needed to use the model. This includes determining how risk can be covered in the model, determining the cost elements of the supply chain, determining where the emissions come from and lastly, what is the location of the vessel.

3.1. Decision making

As explained throughout this report, the decision for a supplier currently is dependent only on cost and risk. However, the research following from this literature review should also include GHG emissions in the decision-making process. To be able to do so, a decision-making tool should help to make the trade-off between the different options based on pre-set parameters that determine the importance of these criteria. The criteria are (as defined in Chapter 2):

- Costs
- GHG emissions
- Risks

The decision-making tool should be able to incorporate these criteria and optimise them simultaneously because there is an interaction between them. However, as explained before, it is hard to implement the risks into the decision-making tool that focuses on implementing GHG emissions. It is already a challenge to properly implement the risk into the maintenance decision (Moussault & Pruyn, 2020). However, part of these risks is mitigated by looking at how far upfront the scheduled job can be completed according to historical data; the delivery of the part should be made before that moment. Therefore, the risks are indirectly accounted for, as explained in Section 3.2.

The decision-making tool should be able to incorporate different supply chain aspects. These aspects entail spare part characteristics (criticality, replacement rate, weight), supplier characteristics (supplier location, manufacturer location), transport requirements (volume, transport mode), delivery requirements (when, where), and operational requirements as set by the decision-maker.

Numerous decision-making techniques to solve this multi-criteria problem are found in the literature. Multi-Criteria Decision-Making (MCDM) (Hsu et al., 2015; Kahraman, 2008; Xiao et al., 2021), Mixed Integer Linear Programming (MILP) (Canales-Bustos et al., 2017; Ghaemi et al., 2022; Liu et al., 2014), or Multi-Objective Evolutionary Algorithms (MOEA) (Cui et al., 2017; Katoch et al., 2021; Xiao et al., 2021; Zhang et al., 2020) are pointed out as suitable algorithms to solve this problem. However, such algorithms are applied to complex solutions with a large solution space. In the case of this thesis, the model only has to consider a reasonable number of solutions, making it possible to find the optimum by calculating each option individually. Based on the data, the model of this thesis will calculate the optimal solution for each spare part separately. But each spare part has only as many options as the decisionmaker would like to consider. As the supplier of the part is fixed and the delivery location is dependent on where the ship has been historically, there is not an infinite solution space.

A technique used a lot in solving multi-criteria problems is Analytic Hierarchy Process (AHP) (Hsu et al., 2015; Kahraman, 2008). AHP can be used as a tool for helping managers structure their problem (of supplier selection) by taking into account all different aspects of the problem (Handfield et al., 2002; Shahroodi & Kambiz, 2012). They also state the real strength of AHP is treating the decision as a system. Even when the problem is new, the structure achieved with an AHP is precious. The AHP is not a substitute for clear human thinking but supports the decision. To create a AHP-model, the decision-maker must still determine the criteria and their respective importance in the decision. Handfield et al. (2002) show how to assess the importance of each criterion by referring to different decision-makers and asking them to rank each criterion compared to another. This outcome weighs all criteria in the eventual AHP-model. On the other hand, Shahroodi and Kambiz (2012) states that the criteria often require data based on knowledge and judgement, which are subjective for the decisionmaker and thus rely on the decision-maker's knowledge of the matter. Another disadvantage, according to Shahroodi and Kambiz (2012) and Tahriri et al. (n.d.), is that AHP does not consider risks and uncertainties in the supplier's performance. This is because AHP considers only the relative performance between alternatives (Yusuff et al., 2001). An Analytic Hierarchy Process (AHP) can be achieved in the model by adding relative weight to one of the criteria or reducing the weight of another. This also makes it fairly easy to implement.

3.2. Alternative spare part demand approach

As explained throughout Sections 2.2 and 2.3, there is not enough data to find correlations or find a standard deviation to the preventive/planned maintenance time when looking at individual jobs. However, within Stolt Tankers, the Planned Maintenance System (PMS) is already adjusted to optimise parts for use as long as possible, so the Mean Time Between Failures (MTBF) is utilised. This means using the available PMS data in combination with the historical data, as suggested by Feng et al. (2021), will already provide a good insight into when the spare parts will be needed. This is done by analysing the available data about the preventive maintenance window of the job and the actual time between the job and the foregoing job.

Based on experience within the case company, it is found that, in practice, on board the vessels, there is a different treatment between the different types of maintenance (L. Teerling, personal communication, May 2023). As discussed before, the two types of maintenance are counter-based and time-based. At the start of the month, the chief engineer looks at the PMS and determines which jobs need to be completed in this month. Therefore, time-based jobs are usually completed before or shortly after (within the same month) the job's due date. This means that it is needed to determine the deviation to the PMS window separately for the two different interval types. This deviation can then be applied to the planning from the PMS, to determine the earliest delivery date to be in time for the job to be carried out.

To perform the analysis of the deviation to the PMS based on historical data, a boxplot on the deviation is created, presenting the distribution of the deviation to the PMS. This deviation is given as a percentage number of the original PMS window, to make it possible to compare the deviations of the different time intervals. When the deviation to the PMS is calculated, it will be possible to have both negative and positive numbers. The negative numbers are when a job is performed before the planned job date, and when the deviation is a positive number, the job is performed after the originally planned date. A boxplot can be used to point out where is the main part of the data and where are the outliers. To be able to say something about the earliest date possible, we have to consider only the lower parts of the boxplot: the lower whisker and the first quartile. When the percentage deviation of the lower whisker is taken into account for planning the earliest delivery date, all deviations (except the outliers) are taken into account. Nonetheless, there are cost benefits associated with using a spare part as close to the planned date as possible, primarily due to the reduced overall spare part consumption during a ship's lifetime. To ensure that the spare part is not kept onboard excessively early, the first quartile of the boxplot is considered as the minimum expected usage date. This covers 75% of the data but also prevents additional costs to the shipping company if this method is implemented.

The only thing to consider is the difference between the different maintenance intervals between jobs. As was seen in Section 2.2.1, the maintenance intervals vary between two months and sixty months or between 300 hours and 36000 hours. When a percentage of these intervals is taken the deviation is way larger in the case of a longer time interval. As the deviation in practice mainly stays within reasonable limits, dependent on the availability of the crew, there is a limit to the percentage deviation. This limit is retrieved by creating a boxplot for the deviation given in days and again taking the first quartile for both interval types.

By applying this new approach, the risks are minimised for the ship, because the delivery is always sufficiently before the estimated job date. Eliminating the risk ensures that the model presented in this research does not have to account for the risks, too.

3.3. Cost of transportation

As said, the decision-making is done based on the costs (Section 3.1). Therefore, this section will establish which costs should be included in the analysis. This means first looking at the costs of procuring a spare part. Secondly, looking at which of these costs can be influenced by the decisionmaker and thus have a variable value to consider for this research.

In the realm of shipment planning, Sahoo et al. (2021) presented a sequence of activities that can be used as a guide to determine the cost of shipping by looking at the price of each step. The sequence as explained by Sahoo et al. (2021) is:
Packing →Warehousing →Trucking →Air transport →Trucking →Warehousing.
An important note is that this sequence is considered for a freight forwarder between the supplier and the receiving party. The whole chain should be considered for this research, which means adding extra layers to the sequence. This is done according to the supply chain layout of the case company, Stolt Tankers.

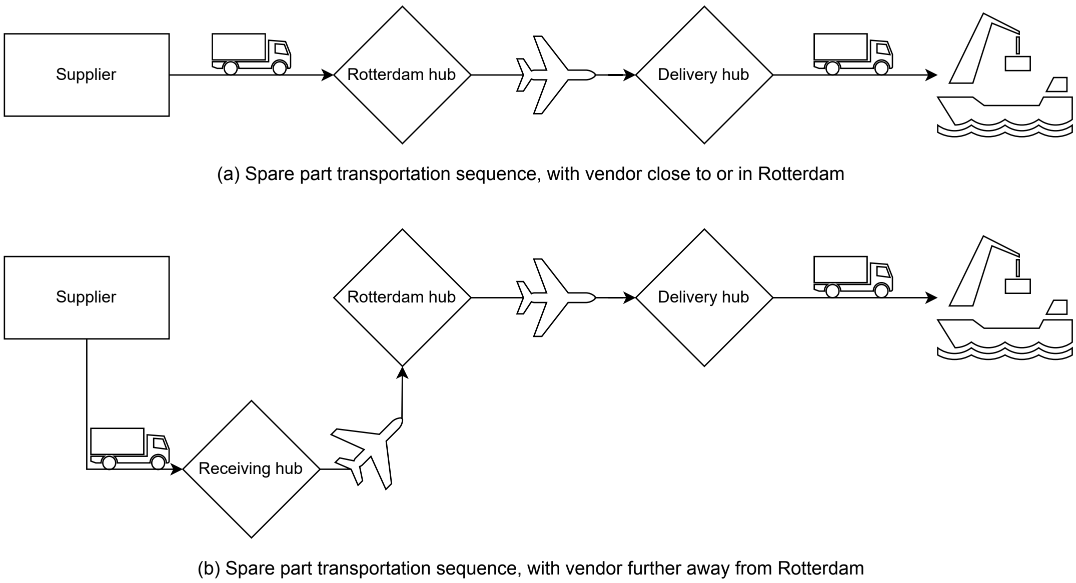


Figure 3.1: Current spare part transportation sequence options Stolt Tankers, with the consolidation of all parts in Rotterdam

According to the author's best knowledge, there has not been much research into optimising spare parts delivery to ships. However, the paper by Vukić et al. (2021) has developed a mathematical model to select the optimal shipment method based on total shipping costs, distance, and delivery time. The three different transportation modes considered by Vukić et al. (2021) are regular, express and mixed variants with air transport. The total cost consists of the sum of the following (Vukić et al., 2021):

- Freight rate of the parcel
- Customs clearance fee
- Airport handling fee
- Logistics operator charge (warehousing)
- Brokerage fee
- Additional charges

The freight rate of the parcel is used to determine the total cost of transport for a parcel. The freight rate in $\$/kg/km$ or $€/kg/km$ can be transferred to the cost as seen in Equation (3.1). The freight rate depends on the route, mode of transport and agreements set with the transport company. The customs clearance fee is paid for each delivery, so when a shipment is consolidated, this cost is made only once. Therefore, it is not considered in the single spare part optimisation considered in this research. The same is valid for the airport handling fee, brokerage fee and possible additional charges (Vukić et al., 2021). The logistics operator is often a company that has their agreements with a company too, therefore it is also really hard to link the costs to one spare part, as it is with the other fees. Hence, the only variable that has a significant impact on the total cost is here the freight rate of a parcel.

$$\text{Cost of Transport} = \text{Freight rate} \times \text{weight} \times \text{distance} \tag{3.1}$$

As can be seen, this only optimises the cost of the delivery, which needs to be extended in this research. Eruguz et al. (2017c) and Pahl (2022) mention that the reason for just-in-time deliveries is valid economic reasons not to have stock on board the vessel. Therefore, early deliveries should be added to the cost equation as the cost of capital. This is calculated according to Equation (3.2), where the cost of capital is given per day. As can be seen, the cost of capital is taken over the full spare part price, so including the cost of transportation. This also includes the price that should be paid for GHG emissions. Multiplying the total cost with the Weighted Average Cost of Capital (WACC), which is a percentage per year, divided by 365.25 (days in a year) gives the cost of capital per day.

$$\text{Cost of Capital} = (\text{Price}_{\text{spare part}} + \text{Freight Cost} + \text{GHG cost}) \times \text{WACC}/365.25 \text{ [euro/day]} \tag{3.2}$$

As the main focus of this research lies in reducing Scope 3 GHG emissions per individual spare part, it will be assumed that all fees will be similar across different airports/trade routes. However, it should be remembered that sending individual packages towards the ships will call for all these additional fees. Also, the emissions will be expressed as a cost to make a decision. This will be elaborated in the next section; Section 3.4

3.4. Emissions

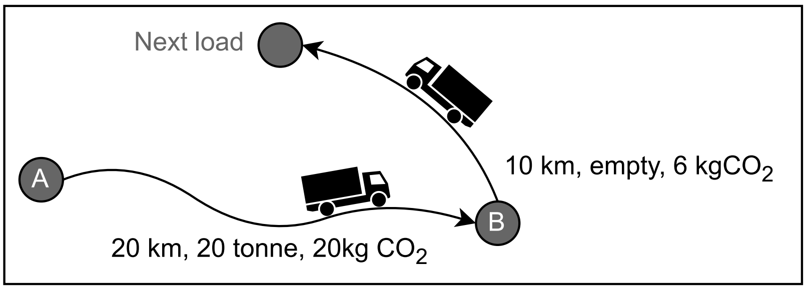
As explained in the introduction of this chapter, for this study, it is necessary to include the GHG emissions in the decision-making process. However, this has not been applied in current literature in the maritime industry. For this research, the emissions considered are limited to those from transporting spare parts, thus not further down the supply chain. Different transport modes can be used to get the spare part from the supplier to the ship, namely aircraft, train, road and ship or a combination of these. For each mode, the amount of GHG emissions produced for a trip over a specific distance is needed. This will be explained in Section 3.4.1. Lastly, to make the trade-off between costs, risks and emissions, the emissions should be converted to a cost, similarly to the risks. This will be described in Section 3.4.2.

3.4.1. Emission factors

To find representative values for GHG emissions, for all transport modes, that can express this for each spare part over each possible distance, Kleijn et al. (2020) published the so-called "**S**tudy on **Tr**ansport **E**missions for **A**ll **M**odes" (*STREAM*). This study provides a comprehensive review of the emissions factors of all freight transport modes, expressed in emissions per ton kilometre, based on 2018 data. The first version of the *STREAM* report was published in 2016. The latest update was performed in 2020: *STREAM* 2020, which presents the key emission factors relevant for climate and air-quality policy-makers. GHG emissions that are included in the report are carbon dioxide (CO_2), methane (CH_4) and nitrous oxide (N_2O). These emissions are expressed as CO_2-equivalent. Next to the greenhouse gasses, also the following pollutants are included in the report: mono-nitrogen oxides

(NO_x), particulate matter (PM_{10}) and sulphur dioxide (SO_2). These air pollutants do not have a direct influence on global warming, and neither do they show a direct effect on the populations in the area of pollution. However, it causes a cumulative effect on air quality problems, such as acid rain (International Maritime Organisation, n.d.-b).

The aim of *STREAM* 2020 is to provide a comprehensive overview of the emission factor for each tonne-kilometre transported by whichever transport mode. This also includes emissions related to the unladen movement of vehicles to move between different transportation jobs; see Figure 3.2 for an explanation and example of a calculation. Kleijn et al. (2020) provides an insight into both exhaust gas emissions (also known as Tank-to-wheel (road & rail) or Tank-to-wake (shipping & aviation) (TTW) emissions) and emissions associated with fuel extraction, production and transport and electricity production and transmission (also known as Well-to-tank (WTT) emissions). Combined, these two calculations give the Well-to-wheel (road & rail) or Well-to-wake (shipping & aviation) (WTW) emissions.



Calculation of CO_2 emission factor per tonne-km (tkm):

- Physical tkm: 20km * 20 tonne + 10km * 0 tonne = 400 tkm;
- CO_2 emissions: 20 kg + 6 kg = 26 kgCO_2;
- Emission factor: 26000/400 = 65 gCO_2/tkm

Figure 3.2: An example of how the emission factor can be calculated for transportation between point A and B, based on Kleijn et al. (2020)

Based on this calculation method, Kleijn et al. (2020) found emission factors for all types of transport. An important note to the values found is that it is based on data retrieved from vehicles or vessels that travelled within, from or to The Netherlands. Examples of representative values are given in Tables 3.1 and 3.2. These values can be used to calculate the pollution of CO_2-equivalents and air pollutants based on the weight of the cargo and the distance travelled. The distance travelled should be considered the route taken, not "as the crow flies". In addition, the *STREAM* 2020 report gives values other than those shown in Tables 3.1 and 3.2 and conversion tables to determine the emission factors when alternative fuels are used.

Table 3.1: Examples of representative emission factors per mode, bulk/packaged cargo transport (Kleijn et al., 2020)

Mode	**Vehicle/Vessel**	**Type of freight**	**CO_2-eq (g/tkm) (WTW)**	**PMc (g/tkm) (TTW)***	**NOx (g/tkm) (TTW)***
Road	Van, empty weight 2,000-2,500 kg	Light	1,326	0.078	4.35
	Truck, medium-size	Med.-weight	256	0.015	1.4
	Tractor-semitrailer, light	Med.-weight	178	0.002	0.53
	Tractor-semitrailer, heavy	Med.-weight	88	0.002	0.22
Rail	Medium-length train (electric 73%: diesel 27%)	Heavy	12	0.001	0.05
Inland shipping	Rhine-Herne canal (RHC) vessel	Heavy	38	0.014	0.4
	Large Rhine vessel	Heavy	24	0.01	0.26
Maritime shipping	Short-sea: General Cargo 10-20 dwkt	Heavy	22	0.009	0.4
	Deep-sea: Bulk carrier 35-60 dwkt	Heavy	6.6	0.003	0.13
Aviation	Long-haul (average)	Light	544	0.015	1.98

* The emission factors for air pollutants provide no indication of the potential health damage associated with the various modes, which depends on where the emissions occur.

To check whether the values given by Kleijn et al. (2020) are representative values for the sector, the

Table 3.2: Examples of representative emission factors per mode, container transport (Kleijn et al., 2020)

Mode	Vehicle/Vessel	Type of freight	CO_2-eq (g/tkm) (WTW)	PMc (g/tkm) (TTW)*	NOx (g/tkm) (TTW)*
Road	Tractor-semitrailer, heavy (2 TEU)	Med.-weight	121	0.003	0.3
Rail	Long train (electric 73%: diesel 27%)	Med.-weight	18	0.0018	0.08
Inland shipping	Rhine-Herne canal (RHC) vessel (96 TEU)	Med.-weight	52	0.019	0.55
	Large Rhine vessel (208 TEU)	Med.-weight	32	0.013	0.34
Maritime shipping	Short-sea: 1,000–1,999 TEU container ship	Med.-weight	32	0.013	0.57
	Deep-sea: 8,000-11,999 TEU container ship	Med.-weight	12	0.005	0.23

* The emission factors for air pollutants do not indicate the potential health damage associated with the various modes, which depends on where the emissions occur.

values are compared to values from other sources. United Nations Framework Convention on Climate (2021) has presented a calculation tool including emission factors for different transport modes. These values are based on data gathered within the United Kindom or used by the UK Government. The values from the transport modes within the tool are similar to those from the values from Kleijn et al. (2020). As this covers the same transport modes as addressed in the report by Kleijn et al. (2020), it can be assumed these values are accurate.
The case company's spare part transportation company base their analysis on the same report. This will make for an easy comparison between a base year and the results of this study, as then the values used for the calculations are the same.

These emission factors can predict the number of emissions produced during a specific trip, with a known weight of goods, see Equation (3.3). This can then be used to compare different trips to each other or to determine the amount of tax that should be paid. The latter is explained in Section 3.4.2.

$$\text{GHG emissions} = \text{Emission factor}_{\text{transport mode}} \times \text{distance}_{\text{transport mode}} \times \text{weight} \tag{3.3}$$

3.4.2. Carbon taxing and emission trading

Literature shows that taxing carbon positively influences companies to reduce their GHG emissions (Li et al., 2021; Ma et al., 2021; Vallés-Giménez & Zárate-Marco, 2020). The European Commission is the first to set up a Emission Trading Scheme (ETS) to regulate CO_2 emissions. The EU ETS works on the 'cap and trade' principle. With a cap, the maximum amount of GHG that can be emitted by the installations covered in a system is limited. This cap can be reduced over time so that the total emissions fall (European Commission, n.d.-a).
At the end of the year, a stocktake will be performed to see how much allowances should be surrendered by an installation to cover its emissions fully. If not enough allowances are paid, heavy fines will be imposed. Suppose an installation has reduced its emissions more than the reduction in available allowances. In that case, it can keep the allowances as spare for the next year or sell them to another installation that is short of allowances (European Commission, n.d.-a).

The European ETS operates in trading phases, which are there to ensure the set climate goals are achieved. Currently, we are in phase 4 (2021-2030), which strengthens the EU ETS to accomplish the goals set in the Paris Agreement. Annually, the overall decline in the total number of emission allowances is accelerated to a rate of 2.2% (compared to 1.74% before 2021) (European Commission, n.d.-d). The EU ETS initially covers the following sectors and gasses (European Commission, n.d.-a):

- Carbon dioxide CO_2 from
  - electricity and heat generation,
  - energy-intensive industries,
  - commercial aviation within the European Economic Area
- Nitrous oxide N_2O from the production of nitric, adipic and glyoxylic acids and glyoxal
- Perfluorocarbons (PFCs) from the production of aluminium.

For these sectors, participation in the EU ETS is mandatory with some exceptions given by European Commission (n.d.-a).

Next, from 2023 onwards, the maritime sector will also be included in the EU ETS. Therefore, maritime companies will also have to pay for their emissions, which will be capped (European Commission, n.d.-c). The price used within the EU ETS can be used to determine the emissions' associated costs. Aviation has already been included in the EU ETS since 2012 (European Commission, n.d.-b). Therefore, the combination of the emissions factors and the price of emissions will make it possible to determine the additional expense for emissions on a specific route. This will make it easier to make a business-relevant decision. The formula to determine the cost is given in Equation (3.4). Here, the price of GHG is given in $€/kgCO_2$-eq

$$\text{GHG cost} = \text{Price of GHG } \times \text{Emission factor}_{\text{transport mode}} \times \text{distance}_{\text{transport mode}} \times \text{weight} \tag{3.4}$$

3.5. Determine location

The delivery location will influence two significant aspects of the decision-making process: 1. The delivery time, closely related to the risk of downtime (Eruguz et al., 2017a; 2017b); and 2. The distance between the supplier and the ship, which influences the amount of greenhouse gasses that are emitted (Kleijn et al., 2020). Hence, knowing where a ship will be located to deliver something onboard is essential. Chemical tankers are considered to have a relatively short planning horizon (Jetlund & Karimi, 2004; Ronen, 1993). This is because chemical tankers are often controlled by so-called tramp operators. There are three types of operation modes for ships according to Jetlund and Karimi (2004) and Ronen (1993). Tramp ships follow the available cargoes and often engage in contracts of affreightments. This makes it hard to make mid-term and long-term plans. Next to that are liner operators, who often control container and general cargo vessels. Liner vessels follow a pre-determined schedule, and thus their planning horizon is relatively long. Lastly, industrial operators own the cargo and reduce transportation costs by transporting in bulk volumes (Pache et al., 2020; Ronen, 1993).

Papers that look into improving the efficiency of a tramp shipping company (environmental or operational improvements) often address the freight rate as the most critical driver for determining where the ship will be in the near future (Dong Mphil, 2022; Lin & Liu, 2011). Even when the planning is determined, the lead time of a port can complicate the delivery and delay the ship's next trip (Pahl, 2022). A conclusion drawn from this is that it remains hard to make a proper planning for tramp ships. This is also because their routes are fixed at the very last moment, which means that deliveries (such as spare parts) may be requested at any of the about 3000 different ports (Mouschoutzi & Ponis, 2022).

This conclusion is also drawn in the literature review performed by Pache et al. (2020). Here is mentioned that a precise classification on the planning horizon of tramp shipping cannot be made. One reason for this is an overlap between the multiple categories of horizons. These classifications and their estimated horizons are listed in Table 3.3. Some operational decisions will result in changes that also affect the tactical decisions and the other way around. Therefore it is indeed hard to quantify the actual planning horizon of a tramp shipping company.

Table 3.3: Classifications of planning horizons for a tramp shipping company, based on (Pache et al., 2020)

Classification	Planning horizon Tramp Shipping
Strategic	Several years
Tactical	Multiple months to a year
Operational	Days to a few weeks

As it will be hard to predict where tramp vessels will be in the future, this report will focus on what could have been saved, looking into historical data. Based on the findings gathered from this research, a value can be given to the possible savings which will lead to alternative prediction methods. Also, in-house information of a company can be used to provide supply chain planners with the necessary information. Also looking at the structure of the supply chain can already narrow down the possible delivery locations. Companies operating worldwide often make use of a distribution hub system. Next to that, it will be easier to deliver to certain ports in the world, due to the company's connections and regulations that apply to that port. These ports are so-called ”ports of convenience” (T. Smolenaars, personal communication, August 2023).

4 Modelling methodology

In Chapters 2 and 3, is explained that for this thesis, a model should be created that can determine the cost and GHG emissions for the procurement of a spare part. This means the model can determine, based on the available data, the distance the spare part has travelled, what the costs were for this transportation and how many GHG emissions were produced. Correspondingly, based on the days between the delivery and the job date, the loss of capital cost should be determined. Then the model should be able to pick alternative delivery locations and check what would be the best option out of all alternatives considered. Because there is a limited amount of solutions, Chapter 3 suggested that the model should perform the calculations for all alternatives instead of creating an algorithm to search the solution space.

This chapter first introduces the model layout by explaining all the steps. Then, a verification is performed on the model to check whether the model performs as expected.

4.1. Model layout

The model created for this research is structured as depicted in Figure 4.1. The dataset as described in Section 2.2 is imported into the model together with the historical location data of the vessel, the location of all hubs that can be used for receiving spare parts and the location at which regular storings to vessels can be performed. The model is created using Python as a coding language. This is used because the open-source software handles data from Excel files well, which is useful with data from different sources within the company. The Python package "pandas" is used to perform column-wise calculations (similar to Excel) (pandas, n.d.).

After importing all data, the model can perform the calculations for each spare part. This is depicted in Figure 4.1 as the yellow arrow indicating all steps performed for the spare part. Also, the indentations and use of new lines help clarify the steps included in each loop. First, the model determines which of the hubs is the closest to the supplier and what distance this has. Then, the deviation to the Planned Maintenance System (PMS) is calculated for each spare part. This is analysed for the complete dataset of spare parts, whereafter, the value can be used to determine the delivery time. Then, for each delivery situation, each WACC and each GHG price, the total costs and emissions are calculated for the original situation and all alternative situations. The decisionmaker also sets the number of alternative situations that are considered. From these options, the option with the lowest cost is considered optimal and will be used in the results. The optimal value is compared to the original situation, which results in the amount of improvement of both cost and emissions.

Throughout this section, the steps of the model are discussed in more depth. Each step is labelled with a number throughout the section corresponding to the number in Figure 4.1. The Python code corresponding to the model is given in Appendix C.

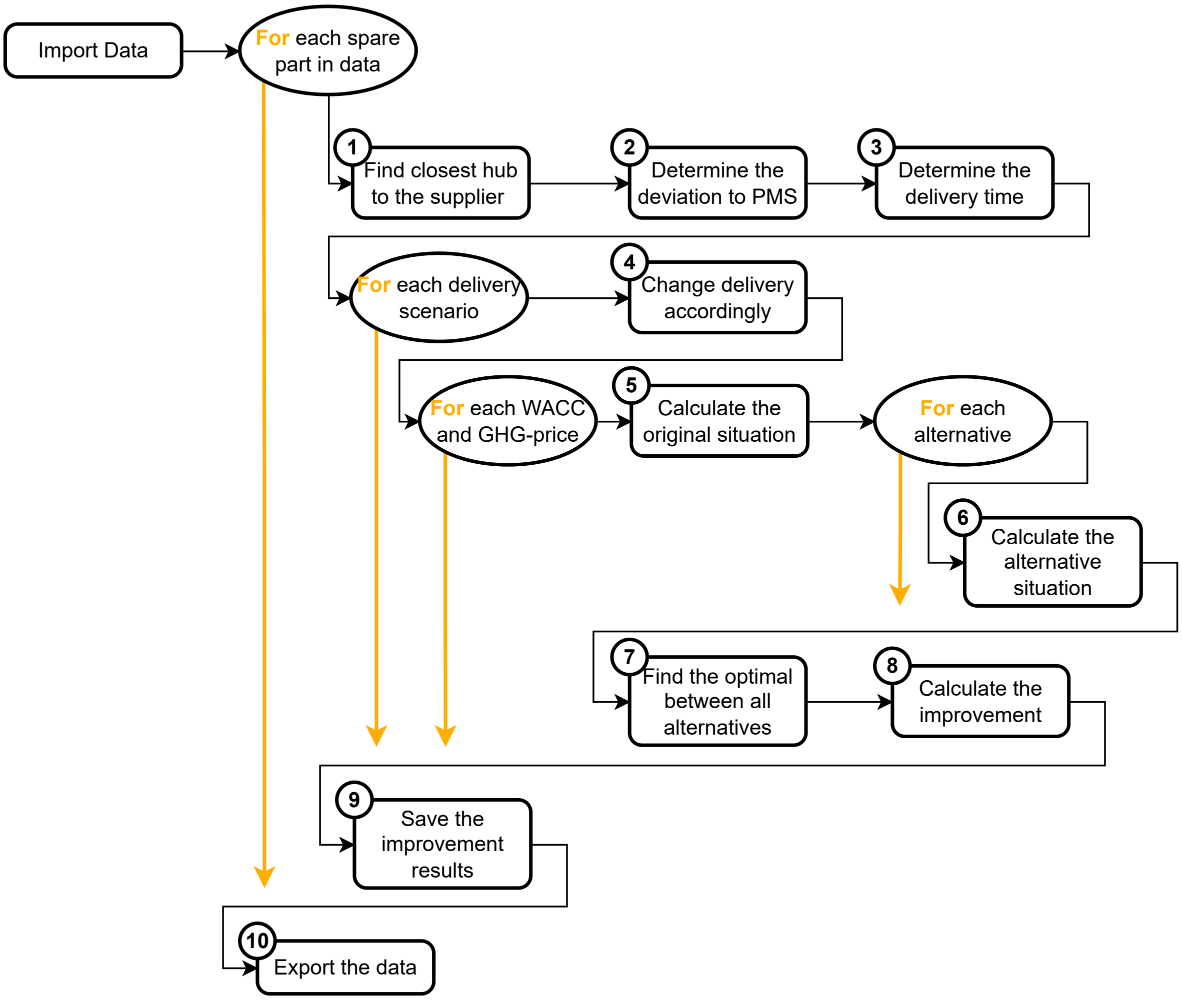


Figure 4.1: Main steps of the designed model

1. Find closest hub to the supplier

In Section 3.5 is explained that when a part is ordered, it will be firstly delivered to a 'receiving hub', which is a hub that the receiving company controls. First, the straight line distance across the earth is calculated between each hub in the list and the supplier of the part. The haversine formula is used for this, as this defines the distance between two points on a sphere. The haversine formula is given in Equation (4.1) (Azdy & Darnis, 2020; Kisanrao Nichat et al., 2013). Here, r is the radius of the earth, equal to 6371km and ϕ and ψ are the latitude and longitude respectively.
After determining the distance for each hub using the haversine formula, the hub with the closest proximity is taken for further analysis.

The next step in the process is calculating the trucking distance, which follows the road between the supplier and the closest hub. The model does so by using the Python package 'requests', which can retrieve data from HTTP protocols. Using an API key from Bing (Microsoft Corporation, n.d.-a), for each trip that should be completed over the road is retrieved using the following HTTP link: https://dev.virtualearth.net/REST/V1/Routes/Driving?wayPoint.1={lat_from},{lon_from}&wayPoint.2={lat_to},{lon_to}&key={api_key} (Microsoft Corporation, n.d.-b). In this link, the 'lat_from', 'lon_from', 'lat_to' and 'lat_to' will be filled in for the supplier (from) and the closest hub (to). The 'api_key' corresponds to an account registered with Bing.

$$\text{Distance} = 2R\sin^{-1}\left(\sqrt{\sin^2\left(\frac{\phi_2-\phi_1}{2}\right)+\cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\psi_2-\psi_1}{2}\right)}\right) \tag{4.1}$$

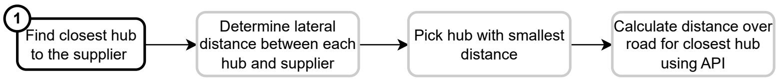


Figure 4.2: Step one in the model: Find closest hub to the supplier

The result of step 1 is the distance between the supplier of the spare part and the receiving hub over the road. This will be used later to determine the trucking cost and the emissions the trip produces.

2. Determine the deviation to PMS

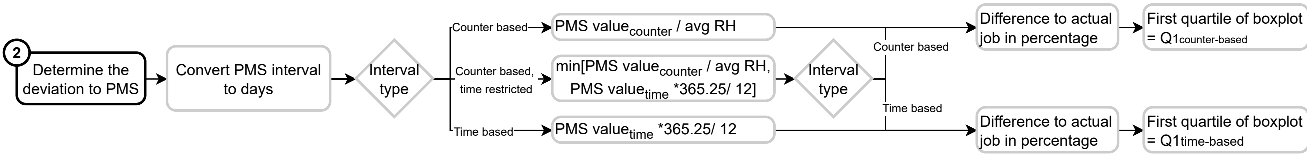


Figure 4.3: Step two in the model: Determine the deviation to PMS

In Section 3.2, it is explained that the deviation to the PMS is calculated and analysed based on historical data to determine the required delivery moment. This step is about performing this analysis and creating the results.

The first step in analysing this data is extracting the PMS interval from the data. There are two types of PMS intervals: 1. Counter-based, which tells the number of running hours before a job should be done, and 2. Time-based, which specifies the exact time between two jobs. Some jobs are counter-based and time-restricted, which means that the job should be done after a specific amount of running hours unless a specific time has passed.

Next, the PMS interval time is converted to days to have the same time format for all interval types: time in days. For counter-based jobs, this is done using the average running hours as specified in Section 2.2, Equations (2.1) and (2.2). After calculating the PM time in actual days, this is compared to the time difference between the current and previous job. This results in a number of days difference between the PM time and how long there was between the jobs, which is converted to a percentage of the PMS window.

Lastly, the first quartile of a boxplot is used to determine a representative value for the PMS deviation without considering the outliers in the dataset. The difference is given both in percentage and in days. As the date difference can be negative (the job is performed earlier than PMS) and positive, taking the first quartile of the boxplot will give a proper insight into the earliest jobs will be performed. This value will then be used in the model's next step to determine the latest delivery time to ensure the spare part is on board before the job starts.

3. Determine the delivery time

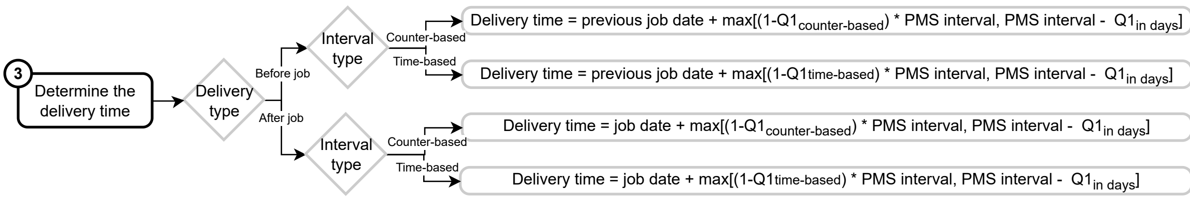


Figure 4.4: Step three in the model: Determine the delivery time

The model's third step is determining the spare part's delivery time. In step 2, the actual PMS window and the deviation to it are given and will be used in this step.
In Section 2.2 is explained that there are two delivery types for jobs. Most jobs order a part after the job, and thus, the delivery is done after the job (and before the next job). This means a stock level should be on board the vessel to perform the job. There are also parts of which there is no stock. These parts should be delivered before the job.

In Figure 4.4, it is shown that a different calculation method is used to determine the maximum delivery date depending on the delivery type and the interval type. For 'before jobs', the previous job is used as a starting point to determine the delivery date. For 'after jobs' the starting point is the current job date. Then, the delivery time is equal to this starting point, added with the PMS interval times one minus the first quartile of the boxplot. The latter makes sure that the spare part would really have been on time for the job. Also, when the PMS times the first quartile is greater than the value given by the first quartile in days, the delivery time is equal to the starting point added with the PMS interval minus the first quartile set in days.

4. Change the delivery time accordingly

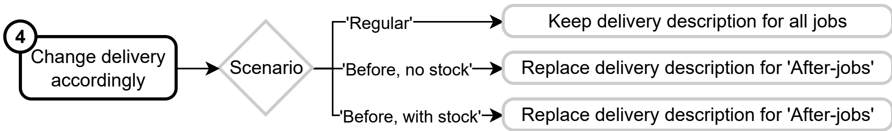


Figure 4.5: Step four in the model: Change the delivery time accordingly to scenario

In Figure 4.5 can be seen that the model can be run for different scenarios. As explained in step 3 of the model (and Section 2.2) in the 'Regular' delivery scenario most spare parts are delivered after the job, with some exceptions; these are delivered before the job. However, from the case company's experience follows that for parts that are delivered after the job, often this delivery is performed as soon as possible, resulting in possible longer transport distances. This raises the idea that delivering the part before the job might be a good decision. This, however, causes the spare part already in stock to remain there and thus there will be an additional capital loss. If this option will improve either GHG emissions, costs, or both should be researched. This scenario is called: 'Before with stock'. To see if the stock has a high influence, there is an additional scenario called: 'Before without stock'. In this scenario, the assumption is made that if the spare part is delivered on time before the job, which should hold using the PMS deviation as described before, no stock is needed.

Another advantage of delivering before the job is that there might be additional delivery options. This is because delivery should be made before the next job on 'after jobs', but the delivery can be earlier than the previous job for the 'before jobs'. This opens the possibility of delivering the part to more different ports, which might improve the reduction of GHG emissions.

5. Calculate the original situation

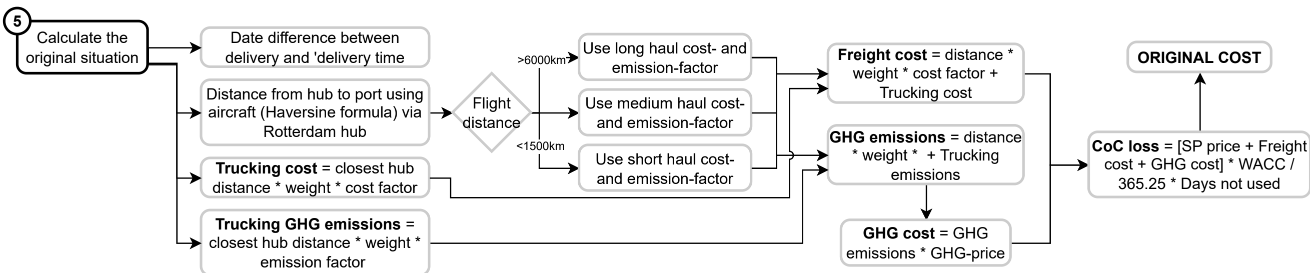


Figure 4.6: Step five in the model: Calculate the original situation

The fifth step in the model is calculating the original situation; see Figure 4.6. The original situation is a representation of the actual delivery within the model. This means the delivery has been made using the Rotterdam hub as an in-between hub to consolidate all orders to the ship. The original situation is calculated using Equations (3.1) to (3.4), which are equations from Chapter 3 repeated below, and Equation (4.2)

First, the trucking cost is calculated using Equation (3.1) taking the weight of the spare part, the freight rate and the distance between the supplier and the closest hub. Using the emissions factor for trucking, the GHG emissions can be calculated according to Equation (3.3). The distance between the supplier and the closest hub is calculated (and explained above) in step 1 of the model.

Then, the flight distance between the hub closest to the supplier and the original delivery port going via the Rotterdam hub is calculated. This is done using the Haversine formula as depicted in Equation (4.1) twice: from the receiving hub to Rotterdam and from Rotterdam to the delivery port. Based on the distance the flight will be classified as short-haul, medium-haul or long-haul according to the following requirements:

- Long-haul flight: > 6000km
- Medium-haul flight: 1500-6000km
- Long-haul flight: < 1500km

For each of the classifications, a different cost- and emission factor should be used to calculate the cost of transport and the GHG emissions for the flight. Using Equations (3.1) and (3.3), the cost and GHG emissions can be calculated for the total flight.
Adding this to the values for the trucking transport gives the total freight cost and total GHG emissions for the spare part. Multiplying the amount of GHG emissions with the cost for GHG per tonne gives the GHG cost.

$$\text{Cost of Transport} = \text{Freight rate} \times \text{weight} \times \text{distance} \tag{3.1}$$

$$\text{GHG emissions} = \text{Emission factor}_{\text{transport mode}} \times \text{distance}_{\text{transport mode}} \times \text{weight} \tag{3.3}$$

$$\text{GHG cost} = \text{Price of GHG } \times \text{Emission factor}_{\text{transport mode}} \times \text{distance}_{\text{transport mode}} \times \text{weight} \tag{3.4}$$

As can be seen in Figure 4.6, only one step remains to find the total original cost for the spare part: calculating the cost of capital that is lost by having the spare part onboard earlier than needed, using Equation (3.2). The model assumes that the spare part is needed at the latest delivery time, calculated in step 3 of the model. Then, the price of the spare part is defined by the price of the spare parts itself, the freight cost and the GHG cost. This is multiplied by the number of days the spare part was delivered before the latest delivery date, times the Weighted Average Cost of Capital (WACC) divided by 365.25. The total cost is then calculated by adding all elements together (Equation (4.2)).

$$\text{Cost of Capital} = \text{Cost}_{\text{spare part}} \times \text{WACC}/365.25 \ [\$/\text{day}] \tag{3.2}$$

$$\text{Total costs} = \text{Spare part price} + \text{Cost of Transport} + \text{GHG emissions} + \text{Cost of Capital} \tag{4.2}$$

6. Calculate the alternative situation

In Figure 4.7, the sixth step of the model is found. This step is performed for each alternative solution. The number of alternative solutions is given as input to the model and is a number telling how many alternative ports should be considered. Based on the historical location data of the ship, the model searches for previous occurrences of the possible delivery ports. For 'Before jobs' it looks at the maximum delivery date and then takes the port calls before that date which are in the list of possible delivery ports. From this list, it takes as many alternatives as defined as input for the model and the model will calculate each of these alternatives.

As can be seen in the Figure 4.7, the calculations are very similar to the calculations for the original situation. There are a few differences that are listed below.

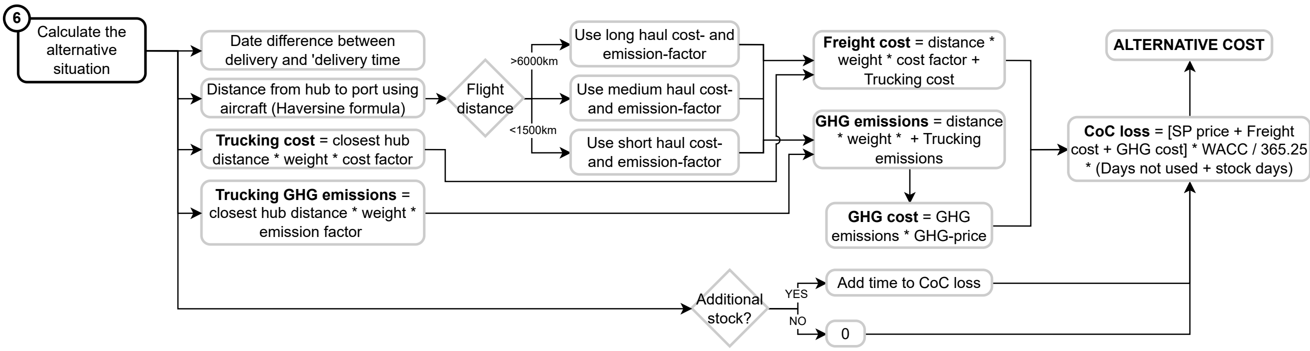


Figure 4.7: Step six in the model: Calculate the alternative situation

1. The distance from the closest hub to the supplier does not go past the Rotterdam hub anymore.
2. Depending on the scenario there is an additional cost of capital loss. This is for the scenario: 'Before with stock'.

In the scenario 'Before with stock' there will be an additional cost of capital loss due to the stock onboard. Figure 4.8 will be used to assist in explaining how the additional cost is calculated. As can also be seen in the picture, for the cost of capital loss, the model considers the time between the delivery and the maximum delivery time. Although the 'Before jobs' look very similar to the 'After jobs', there is a slight difference in the fact that for the 'Before jobs' the delivery is compared to the maximum date before the current job. For 'After jobs' it is compared to the projected maximum delivery date before the next job. As the area between the delivery and the actual job is not predictable and variable, this period is not considered for the cost of capital. To make a fair comparison between the different scenarios, this should also not be considered for the additional cost of capital. Therefore, the additional cost of capital depends on the days between the previous job and the maximum delivery. This is called the 'stock days' in fig. 4.7.

7. Find the optimal between all alternatives

After calculating the total cost for each alternative delivery option, the optimal option should be determined. This is done as shown in Figure 4.9. The model loops through each alternative and saves the option with the lowest cost. It also adds additional columns to the 'pandas' dataframe with the specifics of this option. If there is no alternative available, the original situation is used here, resulting in no improvement later on in the model. It could be that the alternative delivery is the same or very similar to the original situation. In this case, the original situation was already optimal. However, the exact delivery date will most likely always differ from the original as the delivery date in the historical data is not often equal to the arrival date of the vessel in that port. This is only a few days, so the impact is considered negligible.

8. Calculate the improvement

For each attribute listed below, the model will calculate the improvement by subtracting the best alternative from the original situation. If the best option was the original situation, this will return zero. It could be that the most optimal option has a negative improvement in one of the attributes. Then still the overall costs are lower in the alternative, but one (or more) of the attributes performs worse.

Attributes to calculate in step 8:

- Reduction in distance
- Reduction in tonne-kilometre
- Reduction in freight cost
- Reduction in GHG emissions
- Reduction in GHG cost
- Reduction in cost of capital
- Reduction in additional cost of capital
- Reduction in total cost of capital
- Reduction in total cost

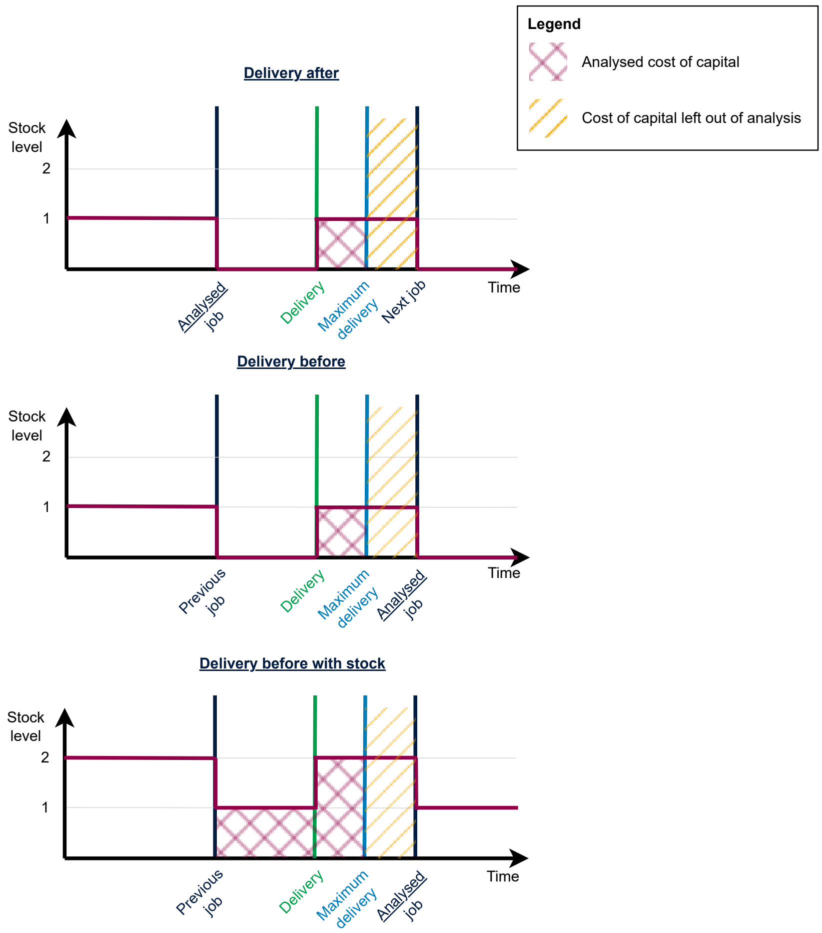


Figure 4.8: Illustrative explanation of the additional cost of capital loss in the 'Before with stock' scenario for an analysed job

9. Save the improvement results

For each of the for loops, meaning for each delivery scenario, each WACC and each cost of GHG emissions, the results are saved as a total of the improvement for each attribute. This means that the reductions as listed above are exported along with the original value, the optimal value and the percentual value with respect to the original situation.

10. Export the data

The last step in the model is to export all the improvement results to Excel. This is done to preserve the data and the results. First, a file (if it does not exist yet) is created using the openpyxl package. Next, using the pandas package, the results are written to the file. Lastly, using the openpyxl package again, filters are added to the headers so the sheet can be easily sorted. Then, the file is saved.

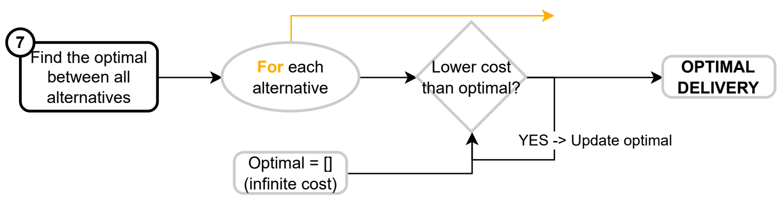


Figure 4.9: Step seven in the model: Find the optimal between all alternatives

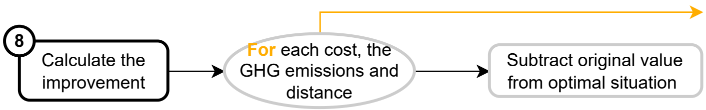


Figure 4.10: Step eight in the model: Calculate the improvement

4.2. Model verification and validation

A model verification is performed to check whether the created model performs the calculations as expected. According to Carson (2002): "Verification occurs when the model developer exercises an apparently correct model for the specific purpose of finding and fixing modelling errors. It refers to the processes and techniques that the model developer uses to assure that his or her model is correct and matches any agreed-upon specifications and assumptions". To distinct verification and validation, it can be added that validation is the process of comparing the result of the model to the actual system. Carson (2002) suggests a simple framework for the verification and validation of the model, consisting of the following steps:

1. Test the model for face validity (verification).
2. Test the model over a range of input parameters (verification).
3. Where applicable, compare model predictions to past performance of the actual system or to a baseline model representing an existing system. When designing a new system, compare implemented model behaviour to assumptions and specifications (validation).

Testing for face validity

Testing the model's face validity means that the model is examined for a given scenario and that the model's outputs are checked for reasonability. The way this model is structured provides a great opportunity to do so for most of the calculation steps in the model. This is because the model performs column-wise operations and creates a column for most steps. Not all data in the dataset is analysed for reasonability, this is only done for a subset selected randomly. Tests performed for the face validity are listed in Table 4.1 and were all successful.

Table 4.1: Face validity tests performed on the model for each step as explained in Section 4.1

Step	Test description
1	Is the selected closest hub indeed the closest? Is the distance between the supplier and the hub reasonable?
2	Is the PMS interval correctly converted based on type? Is the date difference calculated correctly?
3	Is the delivery time calculated correctly, checked manually by following the steps?
4	Are the parts that were labelled 'After' indeed changed to the new delivery method?
5 & 6	Is the distance to the (alternative) port calculated correctly? Are the cost and GHG calculations making sense and are they in the expected range? Are the costs corresponding to the manual calculations?
6	Is the additional cost of capital added correctly?
7	Is the optimal alternative indeed the one with the lowest cost?
8	Is this equal to the manual calculations?

Testing for a range of input parameters

Next to the fact that part of the model will already be run for a range of variables, the model has also been tested on this during the verification tests. In the early stages of the model, the different input parameters were changed to verify if the model was still able to provide results and if the results were as expected. For example, if the WACC is set to zero, will the model only optimise for the cost of transport and the cost of GHG emissions? What was found is that, indeed, if the GHG cost was not zero, the improvement in GHG emissions will reach a fixed value, the highest between all lines with varying WACC values. For other WACC values, the improvement in GHG emissions will grow as the cost for these emissions increases. This means there will be a trade-off for the decision maker whether he or she accepts the loss of capital due to earlier storing or chooses the loss of GHG emissions for a later storing. This also covers the Analytic Hierarchy Process (AHP) part of the model.

Test model behaviour

Carson (2002) mentions that performing a scientific validation is only (theoretically) possible when there is a possibility to match the model to an existing system. This is not the case for this model, as no precise data on the GHG emissions and costs are available. Therefore, it is suggested to look at and validate the model at the micro-level to assess the causes and effects of changes in the model's outcome. This has already been performed quite well during the tests for face validity and testing the range of input parameters. In addition to the previous test, a sample set of the cost results has been presented to an expert in this field within the company (T. Smolenaars, personal communication, August 2023).

Next to the actions described above, validating the model further with the available data will be hard.

4.3. Model usability

The model is created to analyse the historical data available within a shipping company. As is described throughout this chapter, there is quite some data needed to make the analysis. However, the major part of the data is standard data when purchasing spare parts, or when analysing operations of a ship. With regard to retrieving the data, the only challenge that emerged during this thesis was finding the weight of the spare parts. This means that if any company possesses similar data, or at least the parts needed for the model, it will be possible to perform the analysis for any company.

In Section 3.5, for a tramp shipper, it will be hard to make a prediction on the location of the vessel. This is the main reason why the model is not able to be used upfront of the delivery and only looks in hindsight. However, the model can be changed with some minor adjustments such that instead of looking at the historical arrivals in ports of convenience, the decision-maker will be able to fill in multiple options of delivery ports and expected delivery dates to see which would be the best options for all spare parts. For liner vessels, the alternative port arrivals are often set a long time in advance, making a prediction more accurate.

After filling in this information, the model can be run using a fixed value for WACC, GHG cost and delivery scenario according to the decision-makers preference and the company's actuals. Running the model for fixed values and a limited number of alternatives makes the model rather quick, which makes it easy to implement in the daily decision-making of a company. However, next to having information about the future location of the vessel, it also requires proper use of the Planned Maintenance System (PMS), which, therefore, should be up to date.

5

Case study

In this chapter, the model introduced in Chapter 4 is applied to conduct a case study, shedding light on its practical capabilities and accomplishments. The data for this case study is provided by Stolt Tankers. Stolt Tankers is a chemical parcel tanker shipping company, owning 164 ships with 70 deep-sea ships (Stolt Nielsen, n.d.-a). The Stolt Tankers vessel selected for the case study is known internally as ”T0126”. The vessel has proven to have a consistent reporting quality and a global operational footprint, making it a good subject for this case study (see Section 2.2 for a detailed description).

The chapter commences by introducing the essential parameters employed to initialise the model, specifically for the ”T0126” at Stolt Tankers. Subsequently, it will be shown when spare parts must be strategically procured to be on board on time for usage. This is needed because of the difference between the scheduled date of the job and the actual done date, which, in many cases, is before the scheduled date. This outcome serves as a critical component for the model, guiding it for optimal timing for spare part deliveries. Lastly, the results of the model analysis will be elaborated on, providing comprehensive insights into the possible savings of both GHG emissions and money.

5.1. Parameter initialisation

To make the model work, it is initialised by setting parameters as required. This section will briefly introduce the values of these parameters.

Analysed time frame

In Section 4.1, it is explained that the model considers multiple alternative delivery locations corresponding to the historical location data of the vessel. The number of alternatives that should be considered by the model can be specified by the decision maker. Next to the number of alternatives, also a fixed timeframe can be given, in months, to limit the model by time instead of options.

Adhering to current procedures at Stolt Tankers, the model is initiated to analyse for a specific timeframe. This is because spare parts are now still only ordered three to six months in advance. Therefore, the model is executed with a **maximum time frame of 6 months**, covering all options that could have been considered at the time of decision. To analyse the influence of the situation where a spare part is ordered only three months in advance, which is late when looking at the current approach within Stolt Tankers, the model is also initiated with a **maximum time frame of 3 months**. When looking at the planning capabilities for a tramp shipping company, three months is already far on the total planning horizon and therefore, this option also reflects on the outcome when a company has a relatively short planning horizon. With this short planning horizon, it is harder to determine where the ship will be in the future.

Costs

The costs for flight and truck transport are based on internal data within the case company. Based on historical data, the average price over a certain flight route is known and converted to a price per tonne-kilometre, or for the purpose of this study, to a price per kilogram-kilometre. The price for each transport mode is given in Table 5.1 as a factor of the price for trucking transport, denoted with x. The value of x will not be disclosed in this report due to the confidentiality of the number. Different proportions may apply to other cases or companies.

Table 5.1: Cost factors applied in the case study (T. Smolenaars, personal communication, July 2023), where x is a confidential number

Average price for:	**Value**	**Unit**
Plane short-haul	8 x	[$/kg/km]
Plane medium-haul	1.5 x	[$/kg/km]
Plane long-haul	x	[$/kg/km]
Truck, medium size	x	[$/kg/km]

WACC

The Weighted Average Cost of Capital (WACC) calculates the cost of capital due to storing the spare part on board the vessel earlier than needed. The WACC depends on the company's financial situation and the sector it operates in. Therefore, the influence of the WACC on the outcome of the model should be analysed too. Multiple values in the range of the case company's WACC have been analysed to see what it would do to the outcome. Also, a WACC of zero is calculated to show the optimal delivery situation if the cost of capital does not influence the decision. In this case, the delivery time does not influence the outcome, and the result is only optimised for the freight cost and the GHG costs. Values for the Weighted Average Cost of Capital that are used for the case are:

- 0.0%
- 7.0%
- 8.0%
- 9.0%

Emissions factors

In Section 3.4 is explained that emission factors are needed to calculate the GHG emissions as a CO_2-equivalent. The emission factor for aviation as given in Section 3.4 is valid for the average between cargo flights and belly flights on long-distance flights. The Well-to-wheel (road & rail) or Well-to-wake (shipping & aviation) (WTW) $CO_2 - eq$ for all different flights is given in Table 5.2. This table also describes the weighted average between belly-freight and full-freight aircraft. According to Kleijn et al. (2020) the weighted average of full-freight (59%) and belly-freight (41%) is representative of air freight transport. Because the exact flight mode is not known, the weighted average will be taken for this case study.

The emissions factors used for this case study are listed in Table 5.3. As the model knows the weight of the spare part in kilograms, the emission factor from Kleijn et al. is divided by 1000 to go from kg CO_2-eq per tonne-kilometer to kg CO_2-eq per kilogram-kilometer. As can also be seen in the table, but also follows from the model, there are three emission factors for flight transport (based on flight distance) and one for truck transport.

GHG costs

To convert the GHG emissions to a cost, the price of GHG is needed. At the time this report is written, the price for EU carbon permits, which is a measure for the GHG price, is equal €92.14 per tonne CO_2-eq (date: 15/08/2023) (Trading Economics, n.d.). On the same day, the euro and dollar conversion rate is 1.09 (Google Finance, n.d.). In dollars, the carbon permits have a price of $100.72 per tonne.

To be able to say something about the future but also the influence of the carbon price on the outcome of the decision, multiple values are used for this parameter. This is also a way of implementing the Analytic Hierarchy Process (AHP) into the decision-making. If a result with higher carbon prices saves more GHG emissions, but the price is higher, the decision-maker can now make a well-informed decision on what it is worth to save these emissions.
To do so, the model is initialised with the following values for the GHG price, given in $ per tonne CO_2-eq:

Table 5.2: Weighted average Well-to-wheel (road & rail) or Well-to-wake (shipping & aviation) (WTW) emissions for different flight modes [$CO_2 - eq$] (Kleijn et al., 2020)

Aircraft category	Distance range	WTW emissions [$CO_2 - eq$]
Belly-freight aircraft		
Short-haul	<1500 km	910
Medium-haul	1500-6000 km	617
Long-haul	>6000 km	572
Full-freight aircraft		
Short-haul	<1500 km	1399
Medium-haul	1500-6000 km	556
Long-haul	>6000 km	525
Weighted average		
Short-haul	<1500 km	1155
Medium-haul	1500-6000 km	587
Long-haul	>6000 km	549

Table 5.3: Emission factors applied in the case study (Kleijn et al., 2020)

Emission factor for:	Value	Unit
Plane short-haul	1155/1000	[grCO_2-eq/kg/km]
Plane medium-haul	587/1000	[grCO_2-eq/kg/km]
Plane long-haul	549/1000	[grCO_2-eq/kg/km]
Truck, medium size	256/1000	[grCO_2-eq/kg/km]

- 0
- 25
- 50
- 75
- 100
- 125
- 150
- 175
- 200
- 250
- 300
- 350
- 400
- 450
- 500
- 550
- 600
- 650
- 700
- 750
- 800
- 850
- 900
- 950
- 1000

As the model calculates the GHG emissions in kilograms, these values should still be divided by 1000 to be used in the model in the right unit.

Location

As explained in Section 3.5 companies such as the case company Stolt Tankers, operate using distribution hubs. These hubs will function as receiving hubs for the spare parts when they leave the supplier. Some receiving hubs can also function as storage hubs where the ship will eventually receive the spare parts. These hubs are situated near or at the ports of convenience, which for Stolt Tankers are the following:

1. Rotterdam, The Netherlands (NLRTM)
2. Houston, Texas, United States (USHOU)
3. Singapore, Singapore (SGSIN)
4. Antwerp, Belgium (BEANR)
5. Ulsan, South-Korea (KRUSN)
6. Fujairah, United Arab Emirates (AEFJR)
7. Algeciras, Spain (ESALG)
8. Durban, South Africa (ZADUR)

5.2. Deviation to the PM window

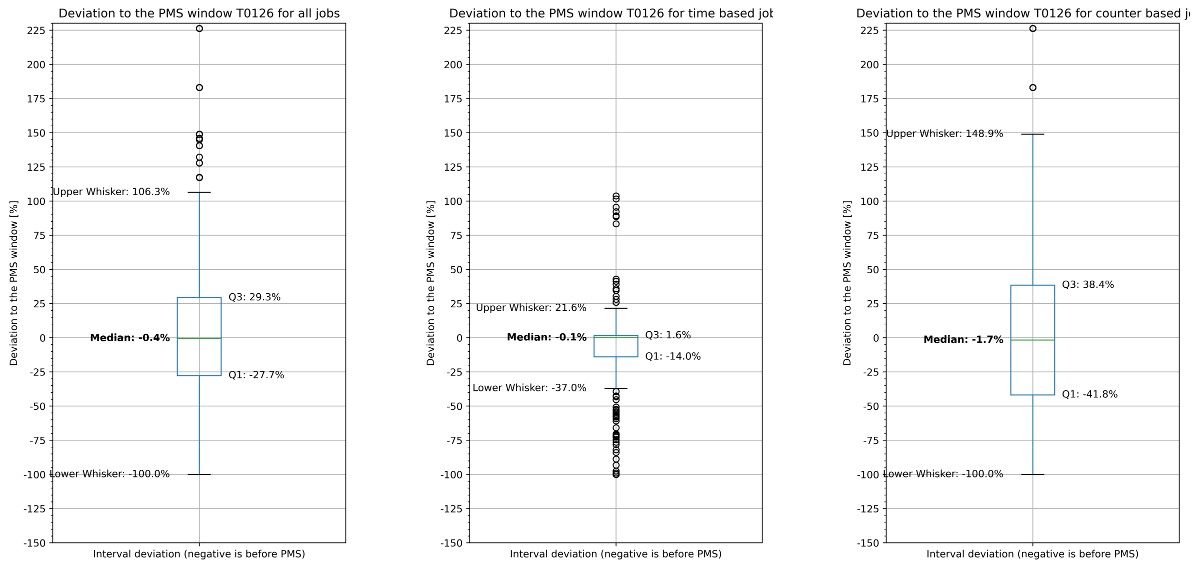
The final parameter yet to be explored is the deviation from the Preventive Maintenance (PM) window. This aspect is an integral part of the model, and this section aims to provide insights into the results regarding the deviation of the Planned Maintenance System (PMS).

In Figure 5.1 the boxplot data is shown for the PMS deviation as a percentage to the original PM window.

As predicted by the knowledge within Stolt Tankers, a difference can indeed be found between time-based and counter-based jobs. For time-based jobs, predicting when a job will be performed is often more accurate than counter-based jobs. This also follows logical reasoning where counter-based jobs depend on the component's running hours and, therefore, are subject to more variation in when the end time is reached. This reasoning is also seen in Figure 5.1c, where the edges of the boxplot (Q1 and Q3) are quite far apart. Correspondingly, the whiskers of the boxplot are quite far out for counter-based jobs, which means that not many points are considered outliers when looking at percentual deviation. For the PMS deviation given in days, see Figure 5.2c, this is different. There are a lot more outliers, which is possibly due to the high amount of jobs with a relatively shorter time period. Even if there is a high percentual deviation, there is still a low deviation in days.

For time-based jobs, in both cases, the percentual deviation (in Figure 5.1b) and the deviation in days (in Figure 5.2b), the boxplots show a larger amount of outliers. This suggests that time-based jobs are often performed near their projected due date, so it is already considered an outlier if that is not the case.

This also proves the necessity that there is a distinction between time-based and counter-based jobs when stating the deviation to the interval as specified in the PMS.



(a) Deviation between the PMS and the job time, for all jobs

(b) Deviation between the PMS and the job time, for time-based jobs

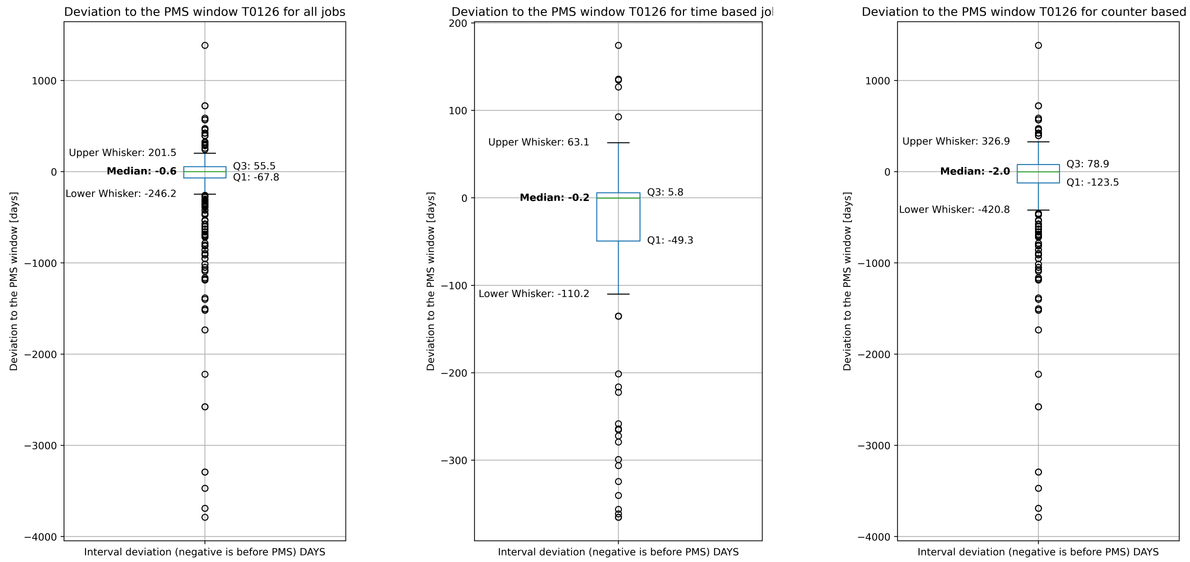
(c) Deviation between the PMS and the job time, for counter-based jobs

Figure 5.1: Comparison of PMS deviation for different job types as a percentage of the original PM window

To present a quantitive summary of the results, Table 5.4 is given. It shows the limits that should be implemented in the model for this case study. These findings provide valuable guidance for optimising spare part delivery, ensuring that they are available on board prior to the PM job, and ultimately making sure the risk of not having the part available is reduced to a very minimum.

Table 5.4: Percentage of the Planned Maintenance System window requiring the spare part to be available on board prior to the Preventive Maintenance job

Interval type	**Deviation to PMS window**	**Maximum deviation days**
Month	*-14.0 %*	*-50*
Counter	*-41.8 %*	*-124*



(a) Deviation between the PMS and the job time, for all jobs

(b) Deviation between the PMS and the job time, for time-based jobs

(c) Deviation between the PMS and the job time, for counter-based jobs

Figure 5.2: Comparison of PMS deviation for different job types **in days**

5.3. Case results

The case study is performed using the parameters as set in Section 5.1, which corresponds to the vessel T0126 within Stolt Tankers. Also, it uses the case results from the analysis of the original delivery time relative to when the part is used, as explained in Section 5.2. This section shows the results of the case study. First, the maximal achievable savings are shown, looking at a time frame of 6 months, which is common within Stolt Tankers. After that, the influence of a lack of information is shown, displaying the results of a 3-month time frame. Next, to see the influence of the Weighted Average Cost of Capital (WACC), it is analysed how changing this value influences the model's behaviour. Now, a fixed value of WACC is used, which is representative of Stolt Tankers, to analyse what the outcome would be if the delivery strategy is changed. The section provides a comprehensive overview of the results, aiding decision-makers in making an informed choice.

The model is run for the dataset as presented earlier. The dataset contains a subset of the spare parts ordered and used within the period of September 21, 2016 to April 27 2023. This is equal to 6.6 years.

5.3.1. Original situation

As explained in Section 4.1, the original costs are calculated the same way as for the alternative situations to ensure a fair comparison between the original situation and the situation chosen by the model. However, the spare part will pass through the Rotterdam hub instead of going from the supplier to the closest hub and then to the hub close to or in the delivery port. In the original situation, the total emissions over the full duration of the time within the dataset is equal to 13.49 tonnes, which equals 2.044 tonnes per year on average (for 6.6 years). All analysed scenarios will be compared to this original situation. The original costs are again marked confidential and are, therefore, not presented in this report. However, there will be an increase/decrease presented as a percentage of the original costs. Important to note is that the original cost of capital and, herewith, the total cost change with the change of the Weighted Average Cost of Capital (WACC). Therefore, for each value of the WACC the original values of these also change and the new situation is compared to this number.

5.3.2. Maximal GHG savings

To assess the maximal savings in GHG emissions, in Section 5.1, it is explained that the model is run using a Weighted Average Cost of Capital (WACC) that is equal to 0%. This ensures the model optimises for the freight cost and the cost of emissions. As these almost linearly follow each other, minimising them together is possible. It should be noted that short-haul flights are more expensive to perform and also produce more GHG emissions per tonne-kilometer. This is why, for a low cost of GHG emissions, the model is expected not yet to reach its maximal potential in reducing GHG emissions.

The result of this test, with a maximum analysed time frame for the alternatives of half a year (6 months), is shown in Figure 5.3. The total maximum saving is equal to 5.85 tonnes (43.4%), which in the graph is shown as 56.6%, for the alternative delivery strategies. The maximal saving is already reached at a GHG cost of 0.5 dollars per kg. However, if the current delivery strategy is used (order after jobs), the maximum saving would be equal to 3.08 tonnes, which is equal to 22.9% savings. When optimising for freight costs only (WACC = 0% and GHG costs = $0.00), it can be concluded that there are fewer savings in GHG emissions (only 21.02% savings), which is in line with not optimising for it. However, a transition point is found after a GHG cost equal to 0.025 $ per kg. After this point, the weight of the GHG cost is considered high enough to influence the outcome of the model so that it will optimise for GHG emissions.

The difference between not taking GHG emissions into account and reaching its maximum potential in all delivery situations equals 1.85%. This means that including the GHG emissions in the decision-making adds a 1.85% saving, compared to not considering it. In Figure 5.4, it can be seen that if there is an increase in the savings of emissions, there is also a slight increase in the internal costs (equal to the sum of the cost of capital and the freight costs) which is equal to 0.03%. Meaning the decision-maker should consider this trade-off. However, this is for a WACC equal to 0%, which has a small influence on the internal cost. In the scenario with internal cost, this influence is expected to be different.

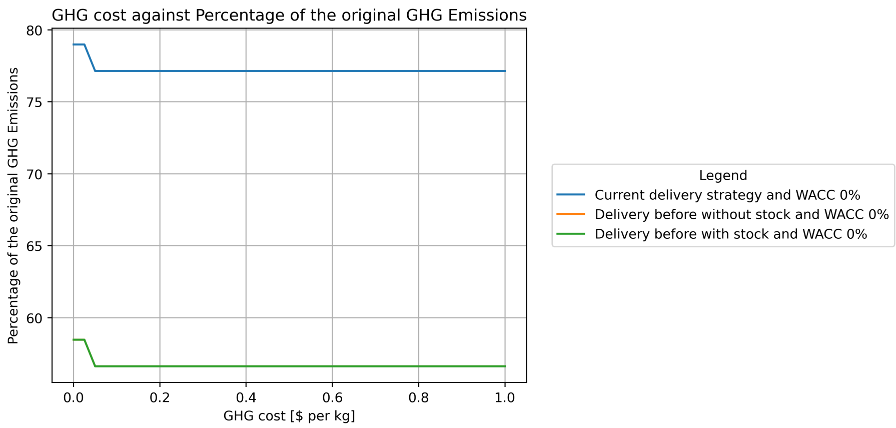


Figure 5.3: Percentual amount of GHG emissions compared to the original situation, leaving out the cost of capital (WACC=0%)

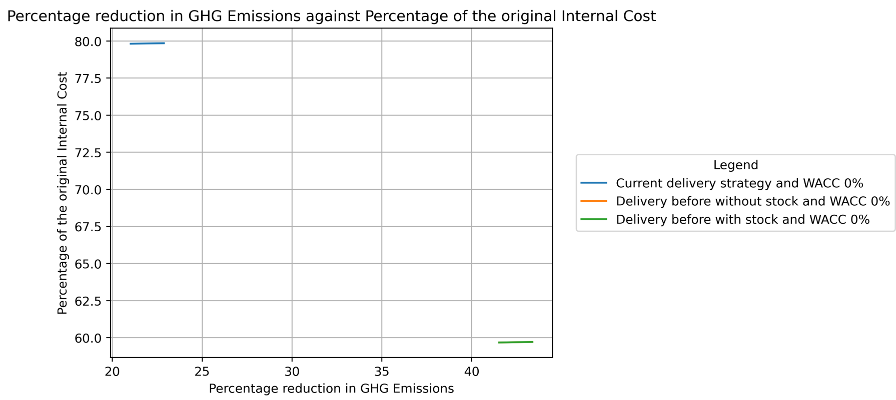


Figure 5.4: Trade-off between the internal cost (freight cost and cost of capital) and the GHG emissions savings for WACC = 0% (more GHG savings, mean higher internal costs)

In Figures 5.3 and 5.4, the lines for 'Delivery before with stock' and 'Delivery before without stock' precisely coincide. This is because when the WACC is set equal to zero, the price of having stock on board is equal to zero. Subsequently, the two scenarios are exactly the same in this case. The same is also valid for the total costs, see Figure 5.5. Here, the total cost consists of the freight cost and the cost of GHG emissions. Each increase in the cost of GHG also decreases the savings in total cost for all delivery strategies. For the strategies where the delivery is performed before the job, this decrease is more per increase in GHG cost than for the current delivery strategy. This is because these two strategies have more savings in GHG emissions compared to the current strategy. If the GHG cost goes up, this becomes a larger part of the total cost, both in the original and in the new situation, and therefore, the percentage savings of the total cost is decreased more in the two alternative strategies.

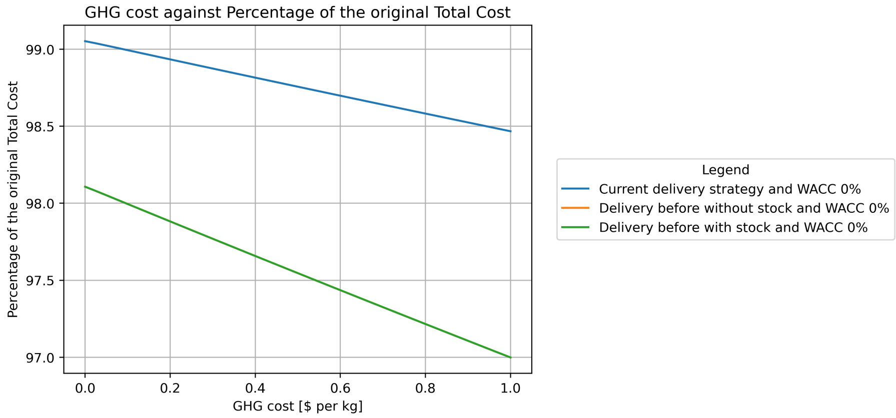


Figure 5.5: Influcence of the GHG cost on the savings in total costs for the three delivery strategies compared to the original situation, leaving out the cost of capital (WACC=0%)

In Figure 5.6, this situation is compared to the original situation based on the delivery locations. This shows that the gross amount of deliveries is done in Rotterdam, The Netherlands, which corresponds to the great number of parts being supplied from The Netherlands (for reference, see Figure 2.5 in Section 2.2.1). In the graph, the port of New Orleans in the United States (USMSY) has zero occurrences in the optimal situation as this is not a port of convenience and thus will not be considered by the model. The port of Durban in South Africa (ZADUR) also has zero occurrences in the optimal situation. This is possibly because there are no suppliers in the neighbourhood of Durban, so the model has chosen another delivery port.

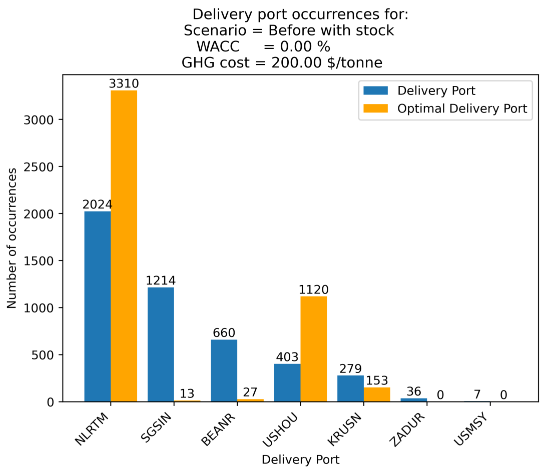


Figure 5.6: Delivery ports in the optimal situation compared to the original situation for maximum GHG savings

5.3.3. Shortening the maximum time frame

In Section 5.1, it is explained that the case company, Stolt Tankers, order their spare parts between three to six months in advance. Therefore, the analysis is also performed for a maximum time frame of 3 months to see the influence of having a more limited planning horizon.

What can be seen in Figure 5.7 is that a similar trend is shown for the results with a large planning horizon of 6 months, but the difference lies with the values. In the results for a planning horizon of 3 months, the percentage of Greenhouse Gas emissions with respect to the original situation is above 100%. In other words, the original deliveries were made emitting less GHG than the analysed situation. Still, this scenario shows the positive effect of including GHG emissions in the decision-making as there is again shown an increase in performance when the GHG cost is increased. However, this increased effect is now reduced to 1.42% for the current delivery strategy and to 1.16% for the other two delivery situations.

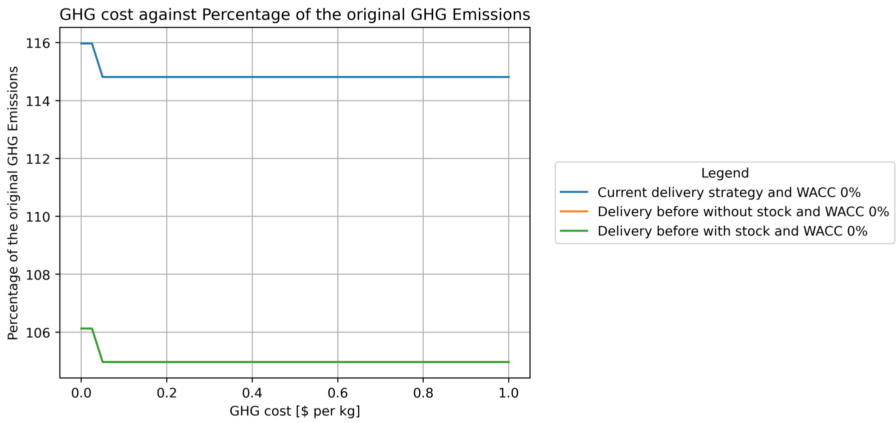


Figure 5.7: Percentual amount of GHG emissions compared to the original situation, leaving out the cost of capital (WACC=0%) for a 3 months time frame

The results of this analysis show the potential for planning ahead for deliveries. By knowing where and when the spare part is needed, there can be more savings in GHG emissions, but also in costs.

5.3.4. Influence of the WACC

Now that the maximum performance of the model is analysed, without considering the cost of capital, the next step is looking at the influence of the Weighted Average Cost of Capital (WACC) on the outcome of the model. It is expected that when the cost of capital increases (a higher WACC), there will be less savings in emissions but higher savings on the cost of capital. This expectation is also found in Figure 5.8 and Figure 5.9, which are almost inverted versions of the other. What can be seen is that when there are more GHG emissions saved (Figure 5.8), the cost of capital increases (Figure 5.9). This also explains the sudden increase in the cost of captial if the savings in GHG emissions increase. At this point in the graph, so at a specific cost of GHG , the cost of emissions is higher than the added cost of capital if the delivery location is changed.

5.3.5. Influence of the delivery strategy

Now that is shown what the influence is of the value for WACC, a reasonable value for the WACC, 8%, within Stolt Tankers is taken to analyse the influence of the applied delivery strategy. As explained, currently most deliveries are made after the job and before the next one (with some exceptions; see Section 2.2). In the other two delivery strategies, the delivery is made before the job, with the idea that this creates more alternative delivery options to be considered. It is already shown in Figure 5.3 that this hypothesis is indeed true. The next step is analysing how this holds when considering the cost of capital.

In Figure 5.10, it can be seen that the current delivery strategy has the lowest percentage in savings of GHG emissions. Both the strategies, where the part is ordered before the job, have the same performance with regards to the GHG emissions. However, because there is a difference in the cost of capital between the two due to the added stock, there is a difference in the cost savings with respect to the original deliveries. This is shown in Figure 5.11, where the savings of total cost is shown as a percentage of the original total cost.

In conclusion, the applied delivery strategy is of influence for the final outcome as such that when one

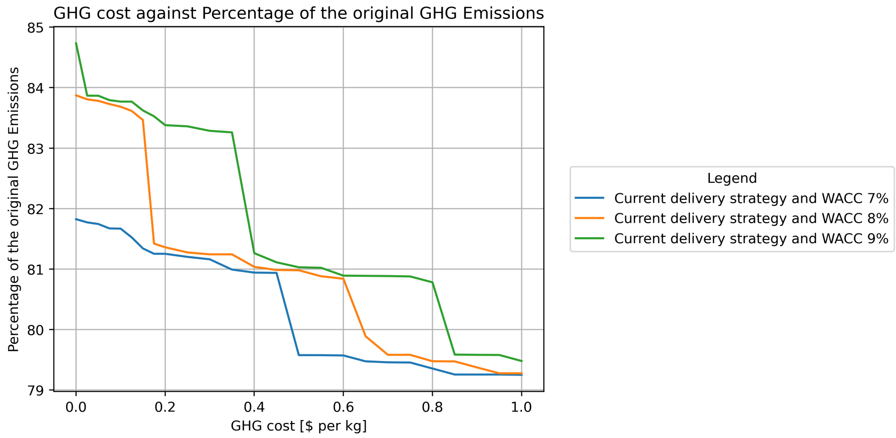


Figure 5.8: Percentual amount of GHG emissions compared to the original situation for the current delivery strategy and varying values of WACC

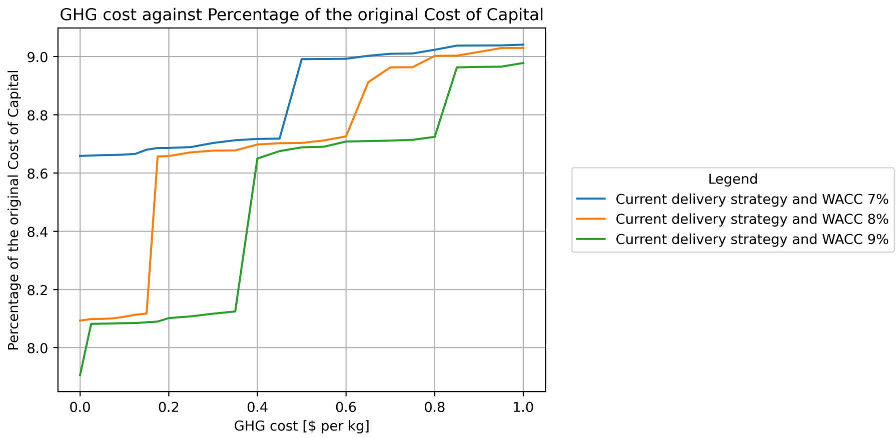


Figure 5.9: Percentual amount of the original cost of capital for the current delivery strategy and varying values of WACC

of the two new strategies is applied, there are more GHG emissions saved. However, up to 0.5 dollars per kilogram of GHG costs, the current delivery strategy outperforms the others based on total cost. In the next subsection, this is more extensively analysed.

5.3.6. Extensive overview of the results

The results presented above are for the specific case of the T0126 from Stolt Tankers. In Section 5.3.2, it is shown that when optimising only for freight cost and GHG cost, already a maximum saving of 22.87% can be achieved when using the current delivery strategy. If the strategy is changed to delivery upfront of the job, the maximum savings increase to 43.38%. However, the Weighted Average Cost of Capital (WACC) should also be included in the decision-making to represent the daily business of any company. The results of this are shown in Sections 5.3.4 and 5.3.5. Regardless, to make the final decision, this section provides additional insight into the decision.

To add to the results as shown in Figure 5.10, it can be seen that the amount of GHG emitted decreases when the cost of GHG emissions becomes higher. When the GHG cost is not considered, which is how Stolt Tankers would currently optimise, the savings in emissions are 35.9% for the delivery strategies where the spare part is delivered before the job. When the current price of GHG is applied, this increases to 36.1%. However, what can be seen in the graph is that when the decision maker adds twice the weight (or the price of GHG will be twice as high), the savings in GHG emissions will rise to 38.4%. This then shows an increase of 2.5% in GHG savings. If the GHG price is increased to $1.0 per kg, the savings in GHG emissions are 40.8%, an increase of 4.8%. When looking at the current delivery strategy, a maximum increase (between a GHG cost of $0 and $1 per kg) equals 4.6%.

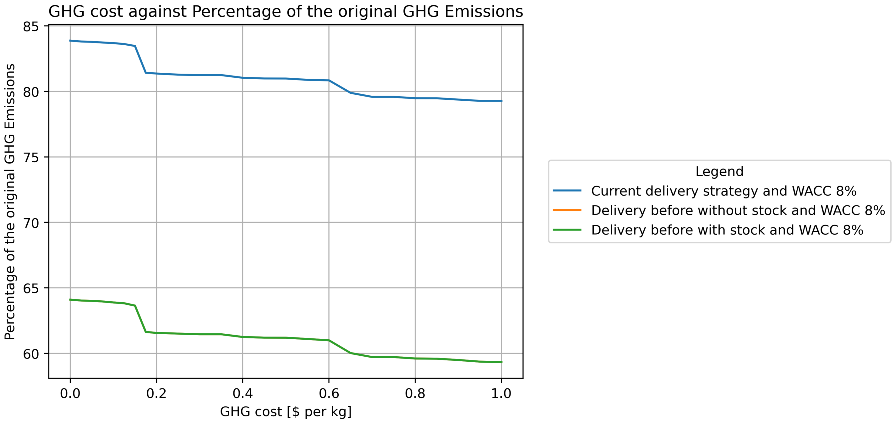


Figure 5.10: Percentual amount of the original GHG emissions for the three delivery strategies when the WACC is 8%

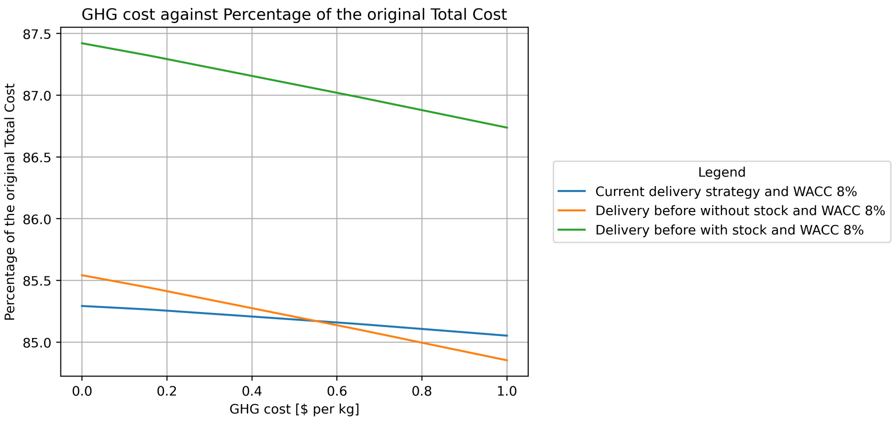
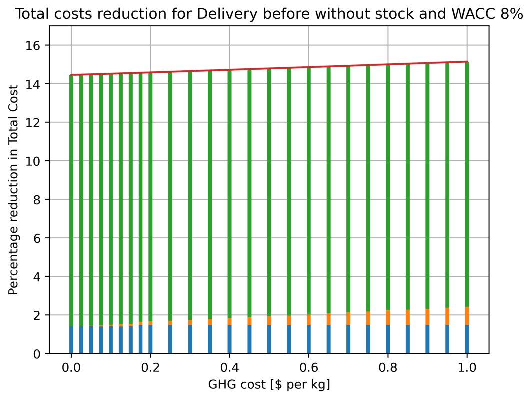


Figure 5.11: Percentual amount of the original total cost for the three delivery strategies when the WACC is 8%

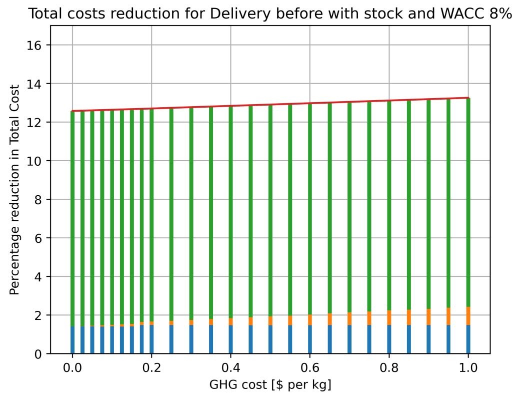
In Figure 5.12, three graphs are shown, providing the breakdown of the specific cost parts with respect to the amount of costs that are saved. Mind that Figure 5.11 is showing a percentage of the original costs and that the graphs in Figure 5.12 are showing the savings. In all graphs in Figure 5.12, there is a high saving in the cost of capital. This means a major part of the total cost savings achieved by the model come from optimising the delivery time with respect to the original situation. It can also be noted from the graphs that in the current delivery strategy (Figure 5.12c), the savings in freight cost and GHG costs are less, which results in a higher saving in the cost of capital. This means there is a trade-off to be made between saving for GHG costs and freight costs or saving in the cost of capital.

The reason for achieving fewer savings with the current delivery strategy is due to the limited delivery options available when deliveries are scheduled between jobs. This limitation potentially leads to reduced savings in both freight costs and GHG emissions. However, it also results in higher savings in the cost of capital. This is primarily because there is less time available between two jobs compared to looking before a job.

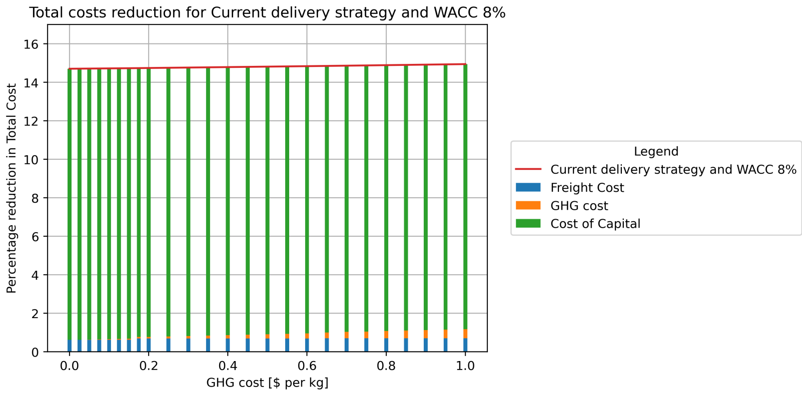
As said, the decision-maker's trade-off will be between the increase in costs and the saving in GHG emissions for that price. This is depicted in Figure 5.13, where the savings in GHG emissions are plotted against the internal costs. These costs consist of the freight costs and the cost of capital. In the plots, it can be seen that a saving in GHG emissions will cost additional internal costs. When looking into Figure 5.14, which shows a more close-up analysis of the strategies with delivery before the job, it can be noted that a 4.77% additional GHG savings cost 0.14% in internal costs. This 0.14% is equal to roughly $650, which equals about $100 on a yearly average. It is up to the decision-maker to maximise both the reduction in internal cost and the reduction in GHG emissions, keeping the company's goals in mind.



(a) The cost savings breakdown for the delivery before the job without stock strategy



(b) The cost savings breakdown for the delivery before the job with stock strategy



(c) The cost savings breakdown for the current delivery strategy

Figure 5.12: Breakdown of the percentage savings in total cost for each of the three delivery strategies, taking a WACC of 8%

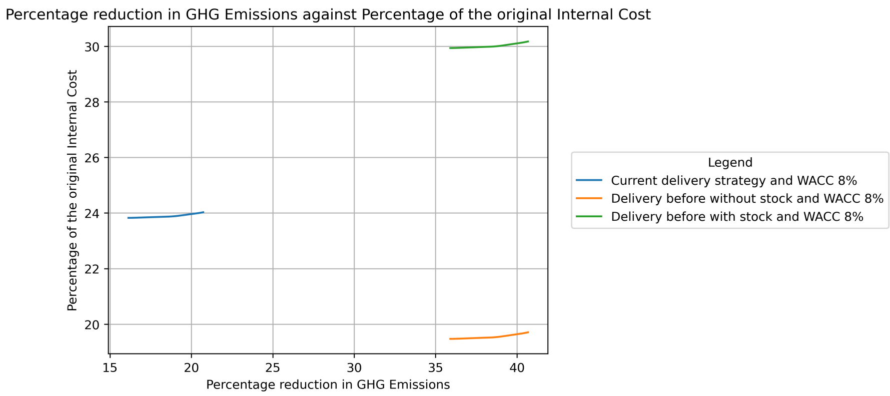


Figure 5.13: Trade-off between the percentage savings in internal cost (freight cost and cost of capital) and the GHG emissions savings for WACC = 8%

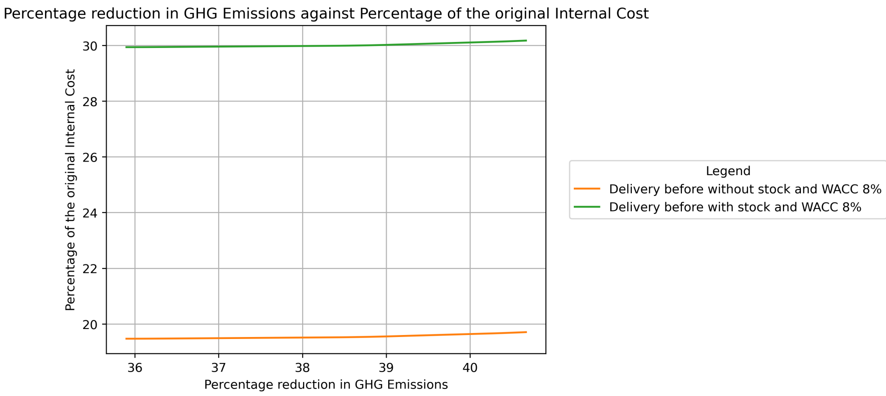


Figure 5.14: Trade-off between the percentage savings in internal cost (freight cost and cost of capital) and the GHG emissions savings for WACC = 8%, for the two before strategies

6 Discussion

This chapter entails the discussion points of this research. Throughout this chapter, the limitations and uncertainties will be discussed. First, the results of the case study are reflected. Next the limitations of the model are pointed out.

6.1. Uncertainties from the data

The data retrieved for the case study relies on some assumptions listed below.

As explained in Section 2.2, the weight of all items within the dataset are added manually based on expert knowledge. Even though this is the highest accuracy available to achieve during the research, this is something to consider. The calculation of the best alternative is based on the trade-off between the cost of capital and the freight cost together with the GHG costs. The latter two are dependent on the weight of the spare part, which is then multiplied by a cost/emission factor and the distance. However, when looking into Figure 6.1, which shows a breakdown of the total cost in the original situation, the freight cost and GHG costs are small compared to the other costs. Therefore, the influence of the weight estimation is expected to be rather small. However, further research should be done to confirm this.

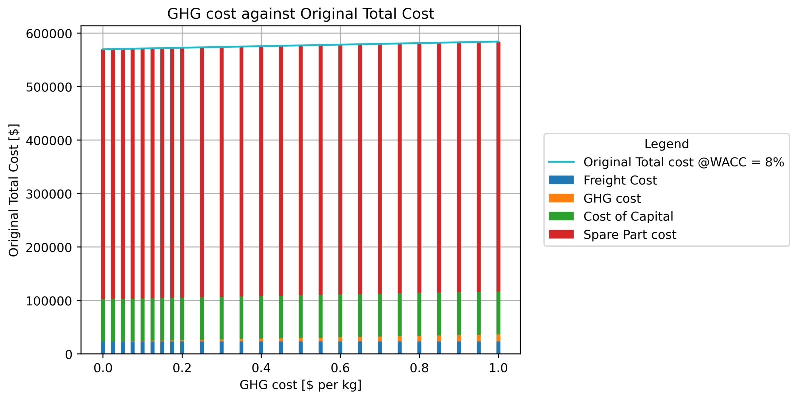


Figure 6.1: Breakdown of the total costs in the original situation

In Section 2.2, it is explained which suppliers are present in the dataset. There is not much variation in the supplier locations, as most of the suppliers are located in The Netherlands. This strengthens the original situation (with respect to GHG savings), where all deliveries are made passing through Rotterdam. However, the intention of this research is to show the potential of changing this delivery strategy and delivering right from the hub that is closest to the supplier. Now, still, most deliveries are made in Rotterdam, which is not showing the effect. The change in outcome is expected to be minimal because the operational profile of the ”T0126” is such that it passes most ports of convenience on a regular basis.

Next to the low variation in suppliers, the data within Stolt Tankers also could not provide alternative supplier locations. A lot of the suppliers get their products from the same factory, only deliver them in a different location. Implementing alternative supplier locations into the model has two possible advantages. The first is that the model can choose different supplier locations based on the vessel location. This makes it possible to minimise both the cost of captial and the freight cost + GHG emissions. The second, more ambiguous, advantage is that this makes it also possible to increase the indirect emissions one step further down the supply chain. However, this also calls for a cultural change on the supplier’s side of the supply chain, which might change the cost aspect for the supplier.

6.2. Model limitations

There are also limitations to the model, mainly parts that were not considered of interest for this thesis. These limitations can be used for further research.

The model does not include the distance between the delivery hub and the ship. In this research, only the distance from the supplier to the first hub and then from this hub to the delivery hub is considered. This is done as the distance between the delivery hub depends highly on the ship’s location. If the ship is at the anchorage, the delivery will be made first using a truck, which bridges the distance between the hub and the barge that performs the delivery. If the ship is at the quay, only a truck will suffice. In both cases, the exact distance is hard to determine. Next to the distance not taken into account, the fees for the arrival port are also not considered. However, there is a difference between all the ports. All this is not included, as the model analyses on a single spare part level, which would increase the model’s complexity. Including this would change the outcome of the model as such that a port further away might be a better choice if the port fees and distance travelled within the port are higher than the difference in total cost between the two ports. This is, therefore, expected only to have a minor influence on the result with respect to the GHG emissions

Next to that, the model now only uses a plane for transportation between the different hubs. This is because the freight forwarder sends almost everything using a plane. However, the trade-off between different transport modes could also influence the performance of the model in a positive way.

As explained in Section 3.1, implementing the risks of not having the spare part delivered on time is rather complicated. The cost of risk consists of multiple attributes, dependent on even more operational characteristics. This is why risks are not taken into account in the model. However, this calls for two limitations. The first is that as an alternative for implementing the risks, the delivery time is adjusted to a deviation from the Planned Maintenance System (PMS). This means that the spare part is possibly delivered earlier to the vessel than really is necessary, and an additional cost of capital might be needed to reflect on the real situation. Nevertheless, this is not taken into account in the model as this part cannot be influenced by the decision as it is defined right now. The second limitation of not including risks is that it is impossible to have a late delivery, according to the model. By including the risks, performing a delivery after the set delivery date implements the cost of risks, resulting in the chance to reflect on this option compared to the others.

Another assumption is that the model considers that all the GHG costs end up at the company that is analysed. However, since the application of carbon taxing is new, the additional costs are not yet determined. In all cases, the model can still be used to assist in decision-making.

6.3. General remarks

The model is applied to the case of a chemical tanker, which operates mainly on the tramp market. As is explained in Section 3.5, a tramp shipper has a short planning horizon. In the results of the case study (see Section 5.3), it has become clear that for a short planning horizon, the amount of GHG savings becomes negative. Meaning that with respect to the original situation, the performance is worse. Furthermore, when the model is run for a larger timeframe than six months, the maximum savings increases even more. However, this is not applicable to the case company and, therefore, is not further addressed in this research. This, however, does mean that the model would even be more applicable for a liner shipper.

7 Conclusion

This last chapter will present the conclusions of this thesis. It will also provide recommendations for further research.

7.1. Conclusion

In this research, a case study is performed to see the influence of taking GHG emissions into consideration for the procurement of spare parts. To find this, the main research question answered in this thesis is:

> *"What is the potential reduction of Greenhouse Gas emissions that is yearly achievable by including indirect emissions into a shipping company's strategy for the logistics of spare parts?"*

To structurally answer the main research question, the six subquestions defined in the introduction (Chapter 1) are answered in consecutive order. The answers to these questions have been presented throughout the report and will be summarised in this chapter to come to a final conclusion for the main research question.

***1.** What does a shipping company's current state-of-the-art supply chain optimisation entail?*

The supply chain consists of 4 elements as described in Section 2.1: Strategies, the network structure, relationships and coordination. As all elements influence each other, the applied maintenance policy also influences the other parts of the supply chain. The most applied maintenance strategy within the maritime industry is Preventive Maintenance (PM). Preventive maintenance is planned maintenance, which makes it possible to analyse fairly easily when a spare part is needed. This makes it possible to use PM for scheduling demand and, therefore, improving the planning. This is why this type of maintenance is analysed in this thesis.

The current supply chain optimisation takes into account the cost and risks. Risk is often considered a function of time and the cost associated with the vessel's downtime. However, because implementing the risk into the thesis complicates the process as it requires a lot of data, the risks for the delivery are minimised throughout this report. Resulting in that this is not taken into account.

***2.** Which modelling approach can be adapted to include Greenhouse Gas emissions in the procurement decision?*

Based on the available data and the application of it, a modelling approach can be chosen. From the available data in Section 2.2, it follows that there is not enough data available to use an algorithm to find the solution to the problem. However, this means that it becomes possible to calculate each solution manually. Criteria for which the decision is made are for the cost and the GHG emissions. From Section 3.1 it follows that the decision making should be done based on the total costs and the GHG emissions. The risks are excluded from the analysis by limiting the risk, as will be explained in the next subquestion.

By adding relative weight between the different criteria (freight cost, cost of capital and GHG costs), an Analytic Hierarchy Process (AHP) can be applied. This means that the decision-maker can influence the outcome of the model. The emissions are included by taking them into account as an additional cost. This cost is determined by the current price for carbon credits and a weight factor that can be applied using the AHP.

3. *When are the spare parts required on-board the vessel based on historical- and preventive maintenance data?*

Currently, within the case company Stolt Tankers and other companies, a Planned Maintenance System (PMS) is used to analyse when maintenance jobs are scheduled in the case of preventive maintenance. However, from experience follows that the jobs are not often performed on the planned date, but somewhere around it. Therefore, this is analysed in the report. Based on the available data, there is not enough available data to apply an algorithm on the deviation to the maintenance window specified by the PMS. Therefore, a statistical analysis is performed using a boxplot. The deviation is given in both a percentage and the actual days. When the value is negative, this shows the number of days the job performed before the actual job date. To provide the model with input about when the spare part is needed, the earliest, the first quartile of the boxplot is taken. This covers 75% of the data point (without outliers) and simultaneously limits the time the job can be performed before the actual job date. This limits the risk of not having the part before the job, and there is still enough time to perform the job before the job date such that it fits the crew's schedule.

4. *How can the chosen modelling approach be applied to sustainable supply chain optimisation in the maritime industry?*

The model has taken shape by including all the elements described above. By including a maximal delivery time for a spare part which is based on the deviation to the PMS, and applying a maximum time frame based on the case company, the model can determine the alternative delivery ports. These alternatives come from historical port calls at ports of convenience. Based on the set parameters and weights (for the Analytic Hierarchy Process (AHP)), the model will determine for all alternatives: the cost of transport, the cost of GHG emissions (and the amount of emissions), the cost of capital and the total cost (which is the sum of the beforementioned and the spare part price). Based on the total costs, the model picks the most cost-efficient option, resulting in an optimised delivery location. The difference between the optimal solution (from the available alternatives) and the original situation is calculated to provide insights about the improvement. This way the model can be used to provide the answer to the main research question.

5. *What is the influence of the decision parameters on the final decision of the model?*

This question has two purposes: to perform the model validation and also to see where the decision maker can have an influence on the outcome of the model. As explained in Section 4.2, next to some verification tests, the different input parameters have been changed to see if the model would behave as expected. Tests have been performed, such as varying the Weighted Average Cost of Capital (WACC) and varying the cost of GHG emissions. This indeed showed the expected behaviour of the model. Furthermore, an analysis is performed on changing the different decision parameters. It is shown that when the timeframe is limited to 3 months instead of 6, the model suggests deliveries that result in a worse performance regarding the GHG emissions with respect to the original situation. Meaning that the set time frame matters for the final outcome of the analysis. Next, the delivery strategy also influences the model results. When looking at the delivery strategies where the spare part is delivered before the job, the model suggests deliveries with less GHG emissions. Both strategies with deliveries before the job perform the same concerning the savings in GHG emissions, however, when looking at the total cost, the delivery without stock outperforms the before delivery with stock. This is because of the reduction in the cost of capital that comes from not having stock onboard the vessel. Still, the current delivery strategy, where almost all parts are ordered (and thus delivered) after the job, outperforms both other strategies up to a GHG cost of $500 per tonne ($0.5 per kg). It is up to the decision maker to decide if the difference in cost savings is worth saving on the GHG emissions and, therefore, choosing another delivery strategy to be applied within the company.

6. *What is the difference between the Greenhouse Gas emissions that have been emitted in the case year and the emissions as analysed by the model in the same year?*

To answer the last subquestion, a case study is conducted on a chemical tanker from Stolt Tankers with an operational profile that goes around the globe. Based on a Weighted Average Cost of Capital (WACC) equal to 8%, which is a representative value in this case, the results have been presented in Section 5.3. Then, dependent on the applied delivery strategy, different amounts of GHG emissions can be saved. It is also shown in the results section that the savings also depend highly on the weight added to the GHG costs. The results for the different delivery strategies are shown in the table below (Table 7.1). The minimal saving in this table presents the savings in GHG emissions when not taking GHG into account in the decision-making. This shows that when optimising for the freight cost and the cost of capital, a significant amount of GHG emissions can already be saved. On a yearly basis, depending on the delivery scenario, the savings in GHG emissions will be 16.13% to 40.67%.

Table 7.1: Results of the GHG savings, for a WACC of 8%, as analysed by the model

Delivery strategy	**Minimum saving ($0 per kg GHG)**	**Maximum saving ($1 per kg GHG)**
Current delivery strategy	16.13%	20.72%
Delivery before without stock	35.90%	40.67%
Delivery before with stock	35.90%	40.67%

This then provides all the information needed to answer the main research question:

"What is the potential reduction of Greenhouse Gas emissions that is yearly achievable by including indirect emissions into a shipping company's strategy for the logistics of spare parts?"

The answer to this question is found in the case of a chemical tanker from Stolt Tankers. This case is an example to show the potential of the applied method and model. As explained above, there are already savings when not optimising for GHG emissions. Therefore, to see the influence of including GHG emissions in the decision-making process depends on the difference between the savings in the situation where GHG is not taken into account and the situation where it is. The result is summarised in Table 7.2.

This shows that changing the delivery strategy only has a 0.2% increase in GHG savings for including GHG emissions. However, when looking into the total savings, changing the delivery strategy does have a significant influence on the total GHG savings. However, when the delivery strategy is changed from the current strategy to a strategy where the delivery is performed before the jobs, this adds an additional 19.77% of savings in emissions.

Table 7.2: Results of the GHG savings, for a WACC of 8%, compared to not including GHG in the decision making

Delivery strategy	**Minimum saving ($0.025 per kg GHG)**	**Maximum saving ($1 per kg GHG)**
Current delivery strategy	0.07%	4.6%
Delivery before without stock	0.07%	4.8%
Delivery before with stock	0.07%	4.8%

In conclusion, including GHG emissions into the decision-making process has a maximum influence of 4.8% compared to when the GHG emissions are not considered. However, this research has also shown the impact of optimising between freight costs and cost of capital on the total GHG emissions. It is evident that when saving on the cost of transport, there is automatically a saving in GHG emissions. This is why including GHG gives only a 4.8% increase in savings. Nevertheless, each percentage point that can be saved can assist in reaching the climate goals set by the governmental organisations, next to a company's own goals.

7.2. Further research

This section presents the recommendations for further research.

The first recommendation is to perform the analysis for more ships and for more spare parts. This will create an even better overview of the capabilities of the model. By taking ships with a less worldwide operational profile or spare parts from different suppliers located more worldwide, the model's outcome might be different.
As already pointed out in the discussion (Chapter 6), taking into account alternative suppliers in the decision will also be interesting because this will make it possible more often to increase both the cost of capital and the GHG emissions at the same time.

Also, in the discussion, it is pointed out that the last part of the trip has not yet been considered within the model. To improve the accuracy of the decision, this could be implemented. The same is valid for the trip definition. Now, only the distance between the supplier and the closest hub is performed using a truck. This means that if the closest hub is Rotterdam, The Netherlands, and the delivery hub is Antwerp, Belgium, the spare part is assumed to be transported to Antwerp using a plane. Implementing this in the model could, for example, be done by setting a threshold for the flight distance.

Lastly, in further research, it could be analysed if the model can be used in the actual procurement of spare parts. This means that the model is adapted as such that when the model is provided with the right information about the ship's expected trips, it can give the best delivery option between the upcoming port calls. The decision maker will then be assisted in deciding where to deliver the spare part. If this is deemed valuable and possible, this is something that could be implemented in the Planned Maintenance System (PMS). This way, decision-makers can enter the different decision parameters, providing them with the right information to make the final decision. However, this requires not only additional research into the applicability, but also a culture change within the company.

Bibliography

Anglou, F. Z., Ponis, S., & Spanos, A. (2021). A machine learning approach to enable bulk orders of critical spare-parts in the shipping industry. *Journal of Industrial Engineering and Management*, *14*(3), 604–621. https://doi.org/10.3926/jiem.3446

Arena, S., Florian, E., Zennaro, I., Orrù, P. F., & Sgarbossa, F. (2022). A novel decision support system for managing predictive maintenance strategies based on machine learning approaches. *Safety Science*, *146*, 105529. https://doi.org/10.1016/J.SSCI.2021.105529

Azdy, R. A., & Darnis, F. (2020). Use of Haversine Formula in Finding Distance Between Temporary Shelter and Waste End Processing Sites You may also like Use of Haversine Formula in Finding Distance Between Temporary Shelter and Waste End Processing Sites. *Journal of Physics: Conference Series*, *1500*, 12104. https://doi.org/10.1088/1742-6596/1500/1/012104

Babaveisi, V., Teimoury, E., Gholamian, M. R., & Rostami-Tabar, B. (2022). Integrated demand forecasting and planning model for repairable spare part: an empirical investigation. *https://doi-org.tudelft.idm.oclc.org/10.1080/00207543.2022.2137596*. https://doi.org/10.1080/00207543.2022.2137596

Canales-Bustos, L., Santibañez-González, E., & Candia-Véjar, A. (2017). A multi-objective optimization model for the design of an effective decarbonized supply chain in mining. *International Journal of Production Economics*, *193*, 449–464. https://doi.org/10.1016/J.IJPE.2017.08.012

Carson, J. (2002). Model verification and validation. *Proceedings of the Winter Simulation Conference*, *1*, 52–58. https://doi.org/10.1109/WSC.2002.1172868

Cassady, C. R., Murdock, W. P., & Pohl, E. A. (2001). Selective maintenance for support equipment involving multiple maintenance actions. *European Journal of Operational Research*, *129*(2), 252–258. https://doi.org/10.1016/S0377-2217(00)00222-8

Cui, Y., Geng, Z., Zhu, Q., & Han, Y. (2017). Review: Multi-objective optimization methods and application in energy saving. *Energy*, *125*, 681–704. https://doi.org/10.1016/J.ENERGY.2017.02.174

Cullum, J., Binns, J., Lonsdale, M., Abbassi, R., & Garaniya, V. (2018). Risk-Based Maintenance Scheduling with application to naval vessels and ships. *Ocean Engineering*, *148*, 476–485. https://doi.org/10.1016/J.OCEANENG.2017.11.044

Dong Mphil, C. (2022). The destination port prediction for tramp ships based on AIS trajectory data mining : a case study of VLCC. https://theses.lib.polyu.edu.hk/handle/200/11999

Eruguz, A. S., Tan, T., & van Houtum, G. J. (2017a). Integrated maintenance and spare part optimization for moving assets. *https://doi-org.tudelft.idm.oclc.org/10.1080/24725854.2017.1312037*, *50*(3), 230–245. https://doi.org/10.1080/24725854.2017.1312037

Eruguz, A. S., Tan, T., & van Houtum, G. J. (2017b). Optimizing usage and maintenance decisions for k-out-of-n systems of moving assets. *Naval Research Logistics (NRL)*, *64*(5), 418–434. https://doi.org/10.1002/NAV.21764

Eruguz, A. S., Tan, T., & van Houtum, G. J. (2017c). A survey of maintenance and service logistics management: Classification and research agenda from a maritime sector perspective. *Computers & Operations Research*, *85*, 184–205. https://doi.org/10.1016/J.COR.2017.03.003

European Commission. (n.d.-a). *EU Emissions Trading System (EU ETS)*. Retrieved *February 15, 2023*, from https://climate.ec.europa.eu/eu-action/eu-emissions-trading-system-eu-ets_en

European Commission. (n.d.-b). *Reducing emissions from aviation*. Retrieved *August 8, 2023*, from https://climate.ec.europa.eu/eu-action/transport-emissions/reducing-emissions-aviation_en

European Commission. (n.d.-c). *Reducing emissions from the shipping sector*. Retrieved *February 17, 2023*, from https://climate.ec.europa.eu/eu-action/transport-emissions/reducing-emissions-shipping-sector_en

European Commission. (n.d.-d). *Revision for phase 4 (2021-2030)*. Retrieved *February 17, 2023*, from https://climate.ec.europa.eu/eu-action/eu-emissions-trading-system-eu-ets/revision-phase-4-2021-2030_en

Feng, Y.-W., Chen, J.-Y., Lu, C., & Zhu, S.-P. (2021). Civil Aircraft Spare Parts Prediction and Configuration Management Techniques: Review and Prospect. *Advances in Mechanical Engineering*, (6), 1–17. https://doi.org/10.1177/16878140211026173

Gandhare, B. S., & Akarte, M. (2012). Maintenance Strategy Selection. *Ninth AIMS International Conference on Management*, 1330–1336. https://www.researchgate.net/publication/267298504

Ghaemi, Z., Tran, T. T., & Smith, A. D. (2022). Comparing classical and metaheuristic methods to optimize multi-objective operation planning of district energy systems considering uncertainties. *Applied Energy*, *321*, 119400. https://doi.org/10.1016/J.APENERGY.2022.119400

Google Finance. (n.d.) *EUR/USD | Google Finance*. Retrieved *August 15, 2023*, from https://www.google.com/finance/quote/EUR-USD?sa=X&ved=2ahUKEwilwc6spd6AAxXL1AIHHdq5AD8QmY0JegQIBhAr

Handfield, R., Walton, S. V., Sroufe, R., & Melnyk, S. A. (2002). Applying environmental criteria to supplier assessment: A study in the application of the Analytical Hierarchy Process. *European Journal of Operational Research*, *141*(1), 70–87. https://doi.org/10.1016/S0377-2217(01)00261-2

Hasni, M., Aguir, M. S., Babai, M. Z., & Jemai, Z. (2018). Spare parts demand forecasting: a review on bootstrapping methods. *https://doi-org.tudelft.idm.oclc.org/10.1080/00207543.2018.1424375*, *57*(15-16), 4791–4804. https://doi.org/10.1080/00207543.2018.1424375

Hertwich, E. G., & Wood, R. (2018). The growing importance of scope 3 greenhouse gas emissions from industry. *Environmental Research Letters*, *13*(10), 104013. https://doi.org/10.1088/1748-9326/AAE19A

Hsu, Y. C., Lu, H. A., & Chu, C. W. (2015). Evaluating and selecting maritime suppliers. *http://dx.doi.org.tudelft.idm.oclc.org/10.1080/03088839.2015.1035351*, *43*(1), 39–58. https://doi.org/10.1080/03088839.2015.1035351

Huiskonen, J. (2001). Maintenance spare parts logistics: Special characteristics and strategic choices. *International Journal of Production Economics*, *71*(1-3), 125–133. https://doi.org/10.1016/S0925-5273(00)00112-2

Ilgin, M. A., & Tunali, S. (2007). Joint optimization of spare parts inventory and maintenance policies using genetic algorithms. *Int J Adv Manuf Technol*, *34*, 594–604. https://doi.org/10.1007/s00170-006-0618-z

International Maritime Organisation. (n.d.-a). *Initial IMO GHG Strategy*. Retrieved *September 22, 2022*, from https://www.imo.org/en/MediaCentre/HotTopics/Pages/Reducing-greenhouse-gas-emissions-from-ships.aspx

International Maritime Organisation. (n.d.-b). *Prevention of Air Pollution from Ships*. Retrieved *February 14, 2023*, from https://www.imo.org/en/OurWork/Environment/Pages/Air-Pollution.aspx

Jetlund, A. S., & Karimi, I. A. (2004). Improving the logistics of multi-compartment chemical tankers. *Computers & Chemical Engineering*, *28*(8), 1267–1283. https://doi.org/10.1016/J.COMPCHEMENG.2003.08.009

Kahraman, C. (2008). *Fuzzy Multi-Criteria Decision Making* (Vol. 16). Springer US. https://doi.org/10.1007/978-0-387-76813-7

Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, *80*(5), 8091–8126. https://doi.org/10.1007/S11042-020-10139-6/FIGURES/8

Kian, R., Bektaş, T., & Ouelhadj, D. (2019). Optimal spare parts management for vessel maintenance scheduling. *Annals of Operations Research*, *272*(1-2), 323–353. https://doi.org/10.1007/S10479-018-2907-Y/FIGURES/11

Kisanrao Nichat, M., RChopde, N., & Nichat, M. K. (2013). Landmark based shortest path detection by using A* Algorithm and Haversine Formula Landmark Based Shortest Path Detection by Using A* and Haversine Formula. *International Journal of Innovative Research in Computer and Communication Engineering*, *1*(2). www.ijircce.com

Kjær, L. L., Pagoropoulos, A., Hauschild, M., Birkved, M., Schmidt, J. H., & McAloone, T. C. (2015). From LCC to LCA Using a Hybrid Input Output Model – A Maritime Case Study. *Procedia CIRP*, *29*, 474–479. https://doi.org/10.1016/J.PROCIR.2015.02.004

Kjaer, L. L., Høst-Madsen, N. K., Schmidt, J. H., & McAloone, T. C. (2015). Application of Environmental Input-Output Analysis for Corporate and Product Environmental Footprints—Learnings from Three Cases. *Sustainability 2015, Vol. 7, Pages 11438-11461*, *7*(9), 11438–11461. https://doi.org/10.3390/SU70911438

Kleijn, A., Hilster, D., Otten, M., & Scholten, P. (2020). *STREAM Goederenvervoer 2020. Emissies van modaliteiten in het goederenvervoer*. Retrieved *February 14, 2023*, from https://ce.nl/publicaties/stream-goederenvervoer-2020/

Lampreia, S., Morgado, T., Navas, H., Cabrita, R., & Requeijo, J. (2022). An Approach to Ship Equipment Maintenance Management. *Lecture Notes in Mechanical Engineering*, 441–450. https://doi.org/10.1007/978-3-030-79165-0{_}41/FIGURES/7

Lazakis, I., Turan, O., & Aksu, S. (2010). Increasing ship operational reliability through the implementation of a holistic maintenance management strategy. *https://doi.org/10.1080/17445302.2010.480899*, *5*(4), 337–357. https://doi.org/10.1080/17445302.2010.480899

Li, P., Lin, Z., Du, H., Feng, T., & Zuo, J. (2021). Do environmental taxes reduce air pollution? Evidence from fossil-fuel power plants in China. *Journal of Environmental Management*, *295*, 113112. https://doi.org/10.1016/J.JENVMAN.2021.113112

Lin, D. Y., & Liu, H. Y. (2011). Combined ship allocation, routing and freight assignment in tramp shipping. *Transportation Research Part E: Logistics and Transportation Review*, *47*(4), 414–431. https://doi.org/10.1016/J.TRE.2010.12.003

Liu, Z., Qiu, T., & Chen, B. (2014). A study of the LCA based biofuel supply chain multi-objective optimization model with multi-conversion paths in China. *Applied Energy*, *126*, 221–234. https://doi.org/10.1016/J.APENERGY.2014.04.001

Ma, Q., Murshed, M., & Khan, Z. (2021). The nexuses between energy investments, technological innovations, emission taxes, and carbon emissions in China. *Energy Policy*, *155*, 112345. https://doi.org/10.1016/J.ENPOL.2021.112345

March, S. T., & Scudder, G. D. (2019). Predictive maintenance: strategic use of IT in manufacturing organizations. *Information Systems Frontiers*, *21*(2), 327–341. https://doi.org/10.1007/S10796-017-9749-Z/FIGURES/9

Microsoft Corporation. (n.d.-a). *Create key - Bing Maps Dev Center*. Retrieved *August 7, 2023*, from https://www.bingmapsportal.com/Application

Microsoft Corporation. (n.d.-b). *Driving Route Example - Bing Maps | Microsoft Learn*. Retrieved *August 7, 2023*, from https://learn.microsoft.com/en-us/bingmaps/rest-services/examples/driving-route-example

Mouschoutzi, M., & Ponis, S. T. (2022). A comprehensive literature review on spare parts logistics management in the maritime industry. *The Asian Journal of Shipping and Logistics*, *38*(2), 71–83. https://doi.org/10.1016/J.AJSL.2021.12.003

Moussault, S., & Pruyn, J. (2020). Modelling Maintenance-Cost Optimised Maintenance in Shipping. https://www.researchgate.net/publication/344670863

Oliver Wyman Forum. (n.d.) *Scope 3 Emissions*. Retrieved *January 10, 2023*, from https://climateactionnavigator.oliverwymanforum.com/scope-3-emissions

Pache, H. ; Grafelmann, M. ; Schwientek, A., Kathrina, ; & Jahn, C. (2020). Tactical planning in tramp shipping - A literature review, 281–308. https://doi.org/10.15480/882.3150

Pahl, J. (2022). Maritime Spare Parts Management: Current State-of-the-Art. *Proceedings of the 55th Hawaii International Conference on System Sciences*. https://hdl.handle.net/10125/79539

pandas. (n.d.) *pandas documentation — pandas 2.1.0 documentation*. Retrieved *August 31, 2023*, from https://pandas.pydata.org/docs/index.html

Pinçe, Ç., Turrini, L., & Meissner, J. (2021). Intermittent demand forecasting for spare parts: A Critical review. *Omega*, *105*, 102513. https://doi.org/10.1016/J.OMEGA.2021.102513

Ranganathan, J., Corbier, L., Schmitz, S., Oren, K., Dawson, B., Spannagle, M., Bp, M. M., Boileau, P., Canada, E., Frederick, R., Vanderborght, B., Thomson, H. F., Kitamura, K., Woo, C. M., Pankhida, N., Miner, R., Pricewaterhousecoopers, L. S., Koch, J., Bhattacharjee, S., … Camobreco, V. (2004). The Greenhouse Gas Protocol.

Ronen, D. (1993). Ship scheduling: The last decade. *European Journal of Operational Research*, *71*(3), 325–333. https://doi.org/10.1016/0377-2217(93)90343-L

Sahoo, R., Bhowmick, B., & Tiwari, M. K. (2021). Developing a model to optimise the cost of consolidated air freight considering the varying scenarios. *https://doi.org/10.1080/13675567.2021.2010682*. https://doi.org/10.1080/13675567.2021.2010682

Schmidt, M., Nill, M., & Scholz, J. (2022). Determining the Scope 3 Emissions of Companies. *Chemical Engineering & Technology*, *45*(7), 1218–1230. https://doi.org/10.1002/CEAT.202200181

Scikit learn. (n.d.) *Choosing the right estimator — scikit-learn 1.2.2 documentation*. Retrieved *June 27, 2023*, from https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Shahroodi & Kambiz. (2012). Application of Analytical Hierarchy Process (AHP) Technique To Evaluate and Selecting Suppliers in an Effective Supply Chain. *Arabian Journal of Business and Management Review*, *1*(1).

Stolt Nielsen. (n.d.-a). *Stolt Tankers | Our businesses | Stolt-Nielsen*. Retrieved *October 10, 2022*, from https://www.stolt-nielsen.com/our-businesses/stolt-tankers/

Stolt Nielsen. (n.d.-b). *Sustainability | Stolt Tankers | Our businesses | Stolt-Nielsen*. Retrieved *October 10, 2022*, from https://www.stolt-nielsen.com/our-businesses/stolt-tankers/sustainability/

Stopford, M. (2008). Maritime economics: Third edition. https://doi.org/10.4324/9780203891742

Tahriri, F., Osman, M. R., Ali, A., & Yusuff, R. M. (n.d.). A REVIEW OF SUPPLIER SELECTION METHODS IN MANU-FACTURING INDUSTRIES. *Suranaree J. Sci. Technol*, *15*(3), 201–208.

Tinga, T. (2013). Principles of Loads and Failure Mechanisms: Applications in Maintenance, Reliability and Design. *Springer Series in Reliability Engineering*.

Trading Economics. (n.d.) *EU Carbon Permits - 2023 Data*. Retrieved *August 15, 2023*, from https://tradingeconomics.com/commodity/carbon

Turan, O., Olcer, A. I., Lazakis, I., Rigo, P., & Caprace, J. D. (2009). Maintenance/repair and production-oriented life cycle cost/earning model for ship structural optimisation during conceptual design stage. *http://dx.doi.org/10.1080/17445300802564220*, *4*(2), 107–125. https://doi.org/10.1080/17445300802564220

United Nations Framework Convention on Climate. (n.d.) *The Paris Agreement*. Retrieved *September 22, 2022*, from https://unfccc.int/process-and-meetings/the-paris-agreement/the-paris-agreement

United Nations Framework Convention on Climate (. 2021). *Greenhouse Gas Emissions Calculator*. Retrieved *March 28, 2023*, from https://unfccc.int/documents/271269?gclid=CjwKCAjwoIqhBhAGEiwArXT7KyHMolS7E-Owe9PE-NrXM_QTNn6zhIbJRJ4JiUoqjeuRnmKsf72fjBoC29sQAvD_BwE

Vallés-Giménez, J., & Zárate-Marco, A. (2020). A Dynamic Spatial Panel of Subnational GHG Emissions: Environmental Effectiveness of Emissions Taxes in Spanish Regions. *Sustainability 2020, Vol. 12, Page 2872*, *12*(7), 2872. https://doi.org/10.3390/SU12072872

Vaughan, T. S. (2005). Failure replacement and preventive maintenance spare parts ordering policy. *European Journal of Operational Research*, *161*(1), 183–190. https://doi.org/10.1016/J.EJOR.2003.06.026

Vukić, L., Stazić, L., Pijaca, M., & Peronja, I. (2021). Modelling the optimal delivery of spare parts to vessels: Comparison of three different scenarios. *http://www.editorialmanager.com/cogenteng*, *8*(1). https://doi.org/10.1080/23311916.2021.1920565

Wang, H. (2002). A survey of maintenance policies of deteriorating systems. *European Journal of Operational Research*, *139*(3), 469–489. https://doi.org/10.1016/S0377-2217(01)00197-7

Wang, W., & Syntetos, A. A. (2011). Spare parts demand: Linking forecasting to equipment maintenance. *Transportation Research Part E: Logistics and Transportation Review*, *47*(6), 1194–1209. https://doi.org/10.1016/J.TRE.2011.04.008

Xiao, B., Liu, C., & Jiang, T. (2021). Multi objective optimization of ship spare parts maintenance based on Improved Genetic Algorithm. *E3S Web of Conferences*, *253*, 02016. https://doi.org/10.1051/E3SCONF/202125302016

Yang, S. K. (2003). A Condition-Based Failure-Prediction and Processing-Scheme for Preventive Maintenance. *IEEE Transactions on Reliability*, *52*(3), 373–383. https://doi.org/10.1109/TR.2003.816402

Yusuff, R. M., Yee, K. P., & Hashmi, M. S. (2001). A preliminary study on the potential use of the analytical hierarchical process (AHP) to predict advanced manufacturing technology (AMT) implementation. *Robotics and Computer-Integrated Manufacturing*, *17*(5), 421–427. https://doi.org/10.1016/S0736-5845(01)00016-3

Zhang, X. W., Liu, H., & Tu, L. P. (2020). A modified particle swarm optimization for multimodal multi-objective optimization. *Engineering Applications of Artificial Intelligence*, *95*, 103905. https://doi.org/10.1016/J.ENGAPPAI.2020.103905

Zhao, J., & Yang, L. (2018). A bi-objective model for vessel emergency maintenance under a condition-based maintenance strategy. *Simulation*, *94*(7), 609–624. https://doi.org/10.1177/0037549717741973/ASSET/IMAGES/LARGE/10.1177{_}0037549717741973-FIG2.JPEG

A

Reporting quality comparison

Table A.1: Comparison of the reporting accuracy of the analysed vessels

Vessel Number	Percentage of parts used	Percentage of unique parts used
T0431	*87.18%*	*39.84%*
T0126	*86.51%*	*55.02%*
T0127	*65.72%*	*38.20%*
T0172	*60.27%*	*131.53%*
T0429	*60.03%*	*26.05%*
T0128	*50.45%*	*34.80%*
T0122	*50.13%*	*30.05%*
T0123	*49.00%*	*23.98%*
T0131	*46.60%*	*36.34%*
T0119	*46.19%*	*43.89%*
T0130	*44.97%*	*25.55%*
T0120	*41.18%*	*35.81%*
T0014	*39.63%*	*30.96%*
T0011	*38.88%*	*35.34%*
T0129	*37.26%*	*30.58%*
T0173	*35.51%*	*41.67%*
T0180	*35.21%*	*41.27%*
T0152	*32.02%*	*39.67%*
T0166	*30.93%*	*35.36%*
T0236	*30.39%*	*30.64%*
T0121	*28.34%*	*35.35%*
T0168	*27.27%*	*46.57%*
T0175	*26.72%*	*36.79%*
T0241	*26.23%*	*28.34%*
T0138	*25.87%*	*31.13%*
T0137	*19.49%*	*20.88%*
T0125	*18.29%*	*23.89%*
T0151	*18.23%*	*33.87%*
T0118	*17.77%*	*24.65%*
T0167	*15.83%*	*25.93%*
T0174	*15.65%*	*30.45%*
T0169	*10.65%*	*20.15%*

B

Data overview

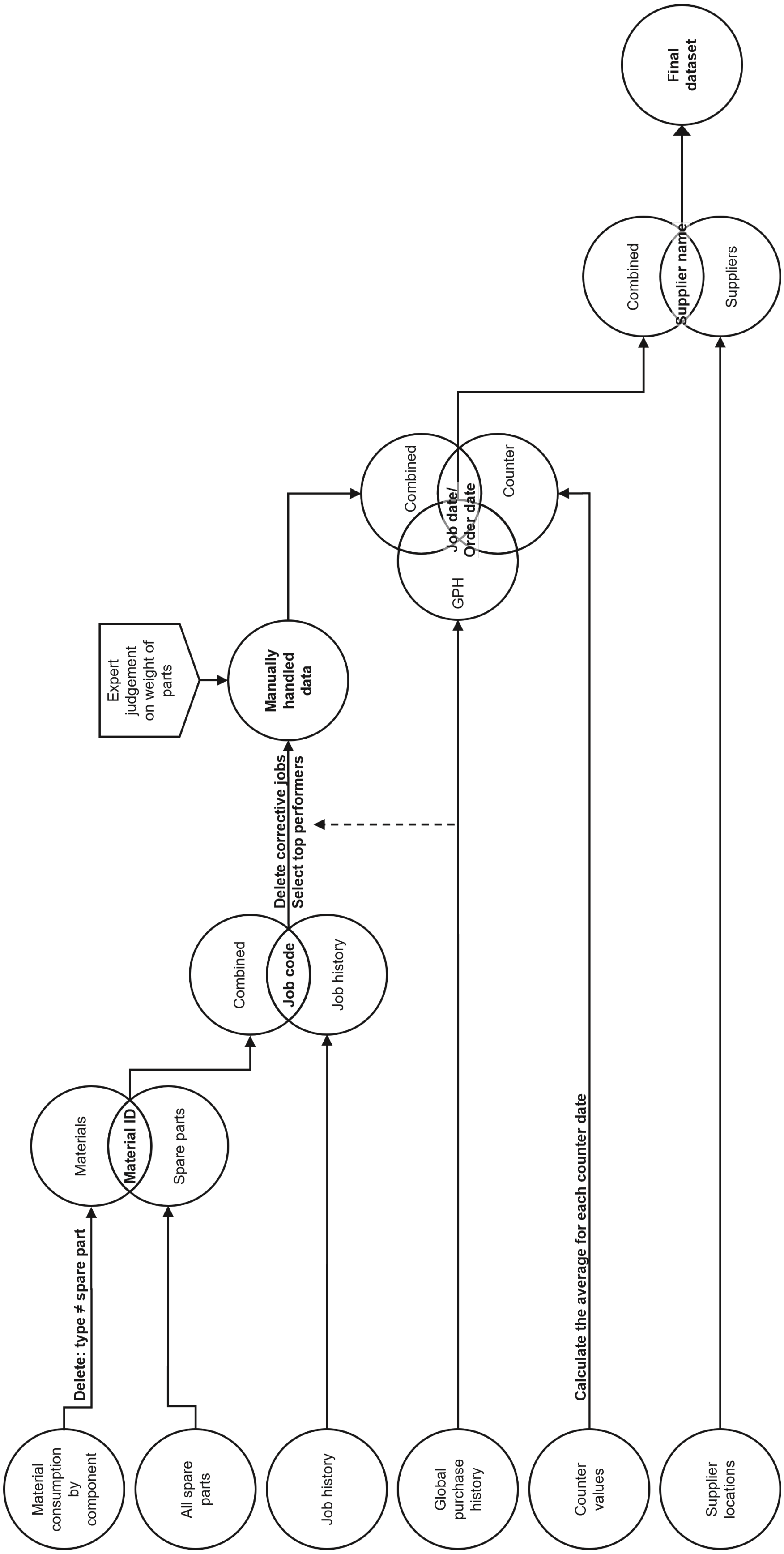


Figure B.1: Overview of how the available data is utilized to form the final dataset

C
Python code of the model

This appendix shows the Python code used as the model.

```python
#%%
# %% Import libraries
import numpy as np
import pandas as pd

import os
import re
import datetime
import shutil
import time
import matplotlib
import math
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import matplotlib.ticker as ticker
from matplotlib.ticker import (MultipleLocator, AutoMinorLocator)
import openpyxl
#Plotly is also a plotting library for Python
import plotly.express as px
from plotly.subplots import make_subplots
from plotly.offline import plot
import plotly.graph_objs as go
import ipywidgets as widgets
from IPython.display import display
import plotly.io as pio
from geopy.geocoders import Nominatim
from geopy.distance import geodesic
from geopy.extra.rate_limiter import RateLimiter
import openrouteservice
import requests

pio.renderers.default='browser'
pd.set_option('display.max_rows', None)
pd.options.mode.chained_assignment = None
plt.rc('axes', axisbelow=True)
plt.close('all')

from warnings import simplefilter
simplefilter(action="ignore", category=pd.errors.PerformanceWarning)

def backup_notebook():
    notebook_file = 'Model_v1.ipynb'  # Replace with your notebook filename
    backup_folder = 'Backups'  # Replace with your backup folder name
    backup_folder_path = os.path.join(os.getcwd(), backup_folder)
    if not os.path.exists(backup_folder_path):
        os.makedirs(backup_folder_path)

    timestamp = datetime.datetime.now().strftime('%Y-%m-%d_%H-%M-%S')
    backup_filename = f'{timestamp}_{notebook_file}'

    shutil.copy(notebook_file, os.path.join(backup_folder_path, backup_filename))
    print(f'Notebook backed up as: {backup_filename}')
#%%
# %% Import all data for ship
path = os.getcwd()
Ship_Name = "Confidential"

# Established dataset
Data_materials = pd.read_excel(os.path.join(path, Ship_Name, f'Final_dataset_{Ship_Name}.xlsx'),sheet_name='Final_dataset')
# Data_materials = Data_materials[Data_materials['Delivery before or after job?'].isin(['Before', '-> Before'])]
Data_materials = Data_materials.sort_values('Delivery Date', ascending=True)

# Historical vessel schedule
Data_vessel_loc = pd.read_excel(os.path.join(path, Ship_Name, f'Vessel schedule all time.xlsx'))
Data_vessel_loc['Arrival Date GMT'] = pd.to_datetime(Data_vessel_loc['Arrival Date GMT'])

# Data for possible delivery ports
Data_ports_or = pd.read_excel(os.path.join(path, f'Unique port occurrences all ships.xlsx'),usecols='A:B')
Data_ports = Data_ports_or[Data_ports_or['Count of deliveries']>=2000]

# Delivery ports from historical vessel schedule
Hist_port_with_del = Data_vessel_loc[Data_vessel_loc['PortCode'].isin(Data_ports['Port'])]

# Unique port data
Port_data = Hist_port_with_del.groupby('PortCode', as_index=False).first()
Port_data = Port_data[['PortCode', 'Port', 'Region', 'Port Country', 'Port Latitude', 'Port Longitude']]

# Data of the Hub locations
Data_hubs = pd.read_excel(os.path.join(path, f'Hub locations (from PDF).xlsx'))
#%%
# %% Here specific parameters are set

## Number of previous ports considered as alternative:
num_of_prev = 1
5 #This is later limited by the date offset
Scenarios = ['Regular', 'Before without stock', 'Before with stock']
# Scenarios = ['Regular']

month_offset = 3
# month_offset = 1200
Date_offset = 365.25/12*month_offset #days of interval used for alternatives


## GHG emissions (divided by 1000 because given in g/tkm)
GHG_plane_sh = 1155 /1000    #grGHG/kg*km based on STEAM2020 CO2-eq, short-haul
GHG_plane_mh = 587 /1000    #grGHG/kg*km based on STEAM2020 CO2-eq, medium-haul
GHG_plane_lh = 549 /1000    #grGHG/kg*km based on STEAM2020 CO2-eq, long-haul

GHG_truck = 256 /1000    #grGHG/kg*km based on STEAM2020 CO2-eq, truck medium size
GHG_van   = 1326/1000    #grGHG/kg*km based on STEAM2020 CO2-eq, van empty weight 2000-2500kg

```

```
## Costs
WACC_ST_vec = np.array([0, 0.07, 0.08, 0.09])

cost_vec1 = np.arange(0, 200, 25)
cost_vec2 = np.arange(200, 1001, 50)
Cost_GHG_perKG_vec = np.concatenate([cost_vec1, cost_vec2])/1000


Cost_flight_sh = Confidential # $/kg/km
Cost_flight_mh = Confidential # $/kg/km
Cost_flight_lh = Confidential # $/kg/km

Cost_flight = Cost_flight_mh

Cost_truck  = Confidential # $/kg/km


## API key
api_key = Confidential
#%%
def parse_interval(interval_str,avg_running_hours):
    try:
        if "/" in interval_str:
            hours_str, months_str = interval_str.split("/")
            hours = int(hours_str[:-1])
            months = int(months_str[:-1])
            return min(datetime.timedelta(hours=hours)*24/(avg_running_hours), datetime.timedelta(days=months*365.24/12)).total_seconds()/(24*60*


        elif interval_str.endswith(("H", "h")):
            return (datetime.timedelta(hours = int(interval_str[:-1]))*24/(avg_running_hours)).total_seconds()/(24*60*60)#*24/(avg_running_hours)
        elif interval_str.endswith(("M", "m")):
            return (datetime.timedelta(days = int(interval_str[:-1])*365.24/12)).total_seconds()/(24*60*60)
        else:
            raise ValueError(f"Unknown interval format: {interval_str}")
    except:
        return np.nan

# Define a function for calculating the distance between two coordinates
def calc_dist_coord(lat_from, lon_from, lat_to, lon_to):
    r_earth = 6371

    #Haversine formula
    dlat = math.radians(lat_to) - math.radians(lat_from)
    dlon = math.radians(lon_to) - math.radians(lon_from)

    distance = 2 * r_earth * math.asin(math.sqrt(math.sin(dlat/2)**2 + math.cos(math.radians(lat_from)) * math.cos(math.radians(lat_to)) * math.s

    return distance

# Define a similar function, but then going from supplier to Rotterdam, Rotterdam to delivery port
def calc_dist_coord_via_hub(lat_from, lon_from, lat_to, lon_to, hub):
    r_earth = 6371

    # hub = 'NLRTM'
    lat_hub = Port_data[Port_data['PortCode'] == hub]['Port Latitude']
    lon_hub = Port_data[Port_data['PortCode'] == hub]['Port Longitude']

    #Haversine formula Supplier to consolidation hub
    dlat1 = math.radians(lat_hub) - math.radians(lat_from)
    dlon1 = math.radians(lon_hub) - math.radians(lon_from)

    distance1 = 2 * r_earth * math.asin(math.sqrt(math.sin(dlat1/2)**2 + math.cos(math.radians(lat_from)) * math.cos(math.radians(lat_hub)) * mat

    #Haversine formula consolidation hub to delivery port
    dlat = math.radians(lat_to) - math.radians(lat_hub)
    dlon = math.radians(lon_to) - math.radians(lon_hub)

    distance2 = 2 * r_earth * math.asin(math.sqrt(math.sin(dlat/2)**2 + math.cos(math.radians(lat_hub)) * math.cos(math.radians(lat_to)) * math.s

    distance = distance1 + distance2
    return distance

def truck_dist(lat_from, lon_from, lat_to, lon_to):
    url = f"https://dev.virtualearth.net/REST/V1/Routes/Driving?wayPoint.1={lat_from},{lon_from}&wayPoint.2={lat_to},{lon_to}&key={api_key}"
    response = requests.get(url)
    data = response.json()

    if 'resourceSets' in data and len(data['resourceSets']) > 0:
        resource = data['resourceSets'][0]['resources'][0]
        distance = resource['travelDistance']
        return distance
    else:
        return None

def find_closest_hub(row):
    min_dist = float('inf')
    closest_hub= None

    for _, hub in Data_hubs.iterrows():
        distance = calc_dist_coord(row['Vendor loc Latitude'], row['Vendor loc Longitude'], hub['Hub loc Latitude'], hub['Hub loc Longitude'])
        if distance < min_dist:
            min_dist = distance
            closest_hub = hub
    return closest_hub

def find_optimal(row):
    lowest_cost = float('inf')  # Initialize with a very high value
    optimal_i = None

    for i in range(1, num_of_prev + 1):
        total_cost = row[f'Total Cost {i}']
        alt_port = row[f'Alternative Port {i}']

```

```
        if alt_port != '':
            if total_cost < lowest_cost:
                lowest_cost = total_cost
                optimal_i = i

    if optimal_i is not None:
        opt_del_date = row[f'Alternative Date {optimal_i}']
        opt_date_diff = row[f'Date difference to (next) job {optimal_i}']
        opt_del_port = row[f'Alternative Port {optimal_i}']
        opt_dist_hub = row[f'Distance Closest Hub to port {optimal_i}']
        opt_fc = row[f'Transportation Freight Cost {optimal_i}']
        opt_ghg = row[f'Transportation GHG {optimal_i} (grams)']
        opt_ghg_cost = row[f'Transportation GHG Cost {optimal_i}']
        opt_coc = row[f'Cost of Capital loss {optimal_i}']
        opt_add_coc = row[f'Additional Cost of Capital loss {i} (stock)']
        opt_coc_noghg = row[f'NO GHG Cost of Capital loss {optimal_i}']
        opt_add_coc_noghg = row[f'NO GHG Additional Cost of Capital loss {i} (stock)']
        opt_tc = row[f'Total Cost {optimal_i}']
    else:
        optimal_i = 'Original'
        opt_del_date = row[f'Original Delivery Date']
        opt_date_diff = row[f'Original Date difference to job']
        opt_del_port = row[f'Original Delivery Port']
        opt_dist_hub = row[f'Original Distance Closest Hub to port']
        opt_fc = row[f'Original Transportation Freight Cost']
        opt_ghg = row[f'Original Transportation GHG (grams)']
        opt_ghg_cost = row[f'Original Transportation GHG Cost']
        opt_coc = row[f'Original Cost of Capital loss']
        opt_add_coc = 0
        opt_coc_noghg = row[f'NO GHG Original Cost of Capital loss']
        opt_add_coc_noghg = 0
        opt_tc = row[f'Original Total Cost']

    return [optimal_i, opt_del_date, opt_date_diff, opt_del_port, opt_dist_hub, opt_fc, opt_ghg, opt_ghg_cost, opt_coc, opt_add_coc, opt_coc_nogh

def calc_delivery_pms_from_prev(row, pms_dev_counter, pms_dev_month):
    if not (pd.isna(row['PMS interval']) or pd.isna(row['Previous Done Since'])):
        if row['Interval Type'] == 'Counter':
            delivery_new = row['Previous Done Since'] + pd.DateOffset(days=max((1-pms_dev_counter) * row['PMS interval'],row['PMS interval']-124)

        elif row['Interval Type'] == 'Month':
            delivery_new = row['Previous Done Since'] + pd.DateOffset(days=max((1-pms_dev_month) * row['PMS interval'], row['PMS interval']-50))
        else:
            delivery_new = row['Done Since']
    else:
        # print(row['PMS interval'])
        # print(row['Previous Done Since'])
        # print('Gaat minder goed')
        delivery_new = row['Done Since']

    return delivery_new

def calc_max_del_from_prev(row, pms_dev_counter, pms_dev_month):
    if not (pd.isna(row['PMS interval']) or pd.isna(row['Done Since'])):
        if row['Interval Type'] == 'Counter':
            delivery_new = row['Done Since'] + pd.DateOffset(days=max((1-pms_dev_counter) * row['PMS interval'],row['PMS interval']-124))

        elif row['Interval Type'] == 'Month':
            delivery_new = row['Done Since'] + pd.DateOffset(days=max((1-pms_dev_month) * row['PMS interval'], row['PMS interval']-50))
        else:
            delivery_new = row['Done Since'] + pd.DateOffset(days=3*365.24)  # if no max is found, then add 3 years (or more if needed)
    else:
        # print(row['PMS interval'])
        # print(row['Previous Done Since'])
        # print('Gaat minder goed')
        delivery_new = row['Done Since']

    return delivery_new


def calc_datediff(row):
    if row['Delivery before or after job?'] in ['Before', '-> After', 'Before without stock', 'Before with stock']:
        return (pd.to_datetime(row['Suggested latest delivery from previous']) - pd.to_datetime(row['Original Delivery Date'])).days
    elif row['Delivery before or after job?'] in ['After', '-> Before']:
        return (pd.to_datetime(row['Max delivery for after jobs']) - pd.to_datetime(row['Original Delivery Date'])).days
    else:
        return None

def calc_cost(distance_plane, distance_truck, weight):
    if distance_plane < 1500:
        fl_c = Cost_flight_sh * distance_plane
    elif (distance_plane >= 1500) & (distance_plane <= 6000):
        fl_c = Cost_flight_mh * distance_plane
    elif distance_plane > 6000:
        fl_c = Cost_flight_lh * distance_plane
    else:
        fl_c = 0

    tr_c = Cost_truck * distance_truck


    return weight * (fl_c + tr_c)

def calc_ghg(distance_plane, distance_truck, weight):
    if distance_plane < 1500:
        ghg_plane = distance_plane * GHG_plane_sh
    elif (distance_plane >= 1500) & (distance_plane <= 6000):
        ghg_plane = distance_plane * GHG_plane_mh
    elif distance_plane > 6000:
        ghg_plane = distance_plane * GHG_plane_lh
    else:
        ghg_plane = 0

    ghg_truck = distance_truck * GHG_truck
```

```
    return weight * (ghg_plane + ghg_truck)

def calculate_percentage(improvement,original):
    if original != 0:
        return (improvement / original) * 100
    else:
        return 0
#%%
# # %% Add the hub that is closed by the supplier
file_path = os.path.join(path, Ship_Name, "Export after closest hub trucking.xlsx")
sheetname = 'Data_materials'

#### Uncomment to redo it. But now it is loaded using pd.read_excel

# Data_materials[['Closest Hub Name', 'Closest Hub Country Name', 'Closest Hub City Name', 'Closest Hub Latitude', 'Closest Hub Longitude']]= Dat
# Data_materials['Straight line Closest Hub distance'] = Data_materials.apply(lambda row: calc_dist_coord(row['Vendor loc Latitude'], row['Vendor
#
# Data_materials['Closest Hub distance'] = Data_materials.apply(lambda row: truck_dist(row['Vendor loc Latitude'], row['Vendor loc Longitude'], r

# if os.path.isfile(file_path):
#     pass
# else:
#     wb = openpyxl.Workbook()
#     wb.active.title = sheetname
#     wb.save(file_path)
#
# with pd.ExcelWriter(file_path, mode="a", engine="openpyxl",if_sheet_exists='replace') as writer:
#     Data_materials.to_excel(writer, sheet_name=sheetname, index=False)
#
# # Load workbook
# wb = openpyxl.load_workbook(file_path)
# for ws in wb.worksheets:
#     ws.auto_filter.ref = ws.dimensions
# wb.save(file_path)


Data_materials = pd.read_excel(file_path)
#%%
if __name__ == '__main__':
    backup_notebook()

Data_materials['PMS interval'] = Data_materials.apply(lambda row: parse_interval(row['Specified Interval Original'], row['Average RH per day Bass

# %% Find time interval between jobs
Data_materials = Data_materials.sort_values(['Material ID', 'Component ID', 'Done Since'])
Data_materials['Days since last replacement'] = Data_materials.groupby(['Material ID', 'Component ID'])['Done Since'].diff().dt.days

Data_materials['Interval deviation (negative is before PMS) DAYS'] = (Data_materials['Days since last replacement'] - Data_materials['PMS interva
Data_materials['Interval deviation (negative is before PMS)'] = (Data_materials['Days since last replacement'] - Data_materials['PMS interval'])/

plot_data_counter = Data_materials[Data_materials['Interval Type']=='Counter']
box_data_counter = plot_data_counter['Interval deviation (negative is before PMS)']
plot_data_month = Data_materials[Data_materials['Interval Type']=='Month']
box_data_month = plot_data_month['Interval deviation (negative is before PMS)']

PMS_dev_counter = -box_data_counter.quantile(0.25)/100
PMS_dev_month = -box_data_month.quantile(0.25)/100


#Determine original values
Data_materials['Original Job date'] = Data_materials['Done Since'] ## This one can be used to plot the done date later in the excel file..
# Data_materials['Suggested delivery maximum'] = pd.NaT
Data_materials['Suggested latest delivery from previous'] = Data_materials.apply(lambda row: calc_delivery_pms_from_prev(row, PMS_dev_counter, PM
Data_materials['Max delivery for after jobs'] = Data_materials.apply(lambda row: calc_max_del_from_prev(row, PMS_dev_counter, PMS_dev_month),axis


Data_materials_SAVE = Data_materials.copy()
Results_DF = pd.DataFrame([])
# Results_V2 = pd.DataFrame(columns=['Scenario', 'GHG cost [$ per kg]', 'WACC', 'Improvement in Distance', 'Improvement in tkm', 'Improvement in

Results_V2 = pd.DataFrame(columns=[
                'Scenario',
                'GHG cost [$ per kg]',
                'WACC',
                'Reduction in Distance [km]',
                'Reduction in tonne-kilometer [tkm]',
                'Reduction in Freight Cost [$]',
                'Reduction in GHG Emissions [t]',
                'Reduction in GHG Cost [$]',
                'Reduction in Cost of Capital [$]',
                'Reduction in additional Cost of Capital [$]',
                'Reduction in Total Cost of Capital [$]',
                'NO GHG Reduction in Cost of Capital [$]',
                'NO GHG Reduction in additional Cost of Capital [$]',
                'NO GHG Reduction in Total Cost of Capital [$]',
                'Reduction in Total Cost [$]',
                'Reduction in Internal Cost [$]',
                'Original Freight Cost [$]',
                'Original GHG Emissions [t]' ,
                'Original GHG Cost [$]' ,
                'Original Cost of Capital [$]',
                'Original additional Cost of Capital (stock days) [$]',
                'Original Total Cost of Capital [$]',
                'NO GHG Original Cost of Capital [$]',
                'NO GHG Original additional Cost of Capital (stock days) [$]',
                'NO GHG Original Total Cost of Capital [$]',
                'Original Total Cost [$]',
                'Original Internal Cost [$]',
                'Optimal Freight Cost [$]',
                'Optimal GHG Emissions [t]' ,
                'Optimal GHG Cost [$]' ,
                'Optimal Cost of Capital [$]',
                'Optimal additional Cost of Capital (stock days) [$]',
```

```
                'Optimal Total Cost of Capital [$]',
                'NO GHG Optimal Cost of Capital [$]',
                'NO GHG Optimal additional Cost of Capital (stock days) [$]',
                'NO GHG Optimal Total Cost of Capital [$]',
                'Optimal Total Cost [$]',
                'Optimal Internal Cost [$]',
                'Percentage reduction in Freight Cost',
                'Percentage reduction in GHG Emissions',
                'Percentage reduction in GHG Cost',
                'Percentage reduction in Cost of Capital',
                'Percentage reduction in additional Cost of Capital',
                'Percentage reduction in Total Cost of Capital',
                'NO GHG Percentage reduction in Cost of Capital',
                'NO GHG Percentage reduction in additional Cost of Capital',
                'NO GHG Percentage reduction in Total Cost of Capital',
                'Percentage reduction in Total Cost',
                'Percentage reduction in Internal Cost'
            ])


max_num_alts = float('-inf')
for scenario in Scenarios:
    for wacc in WACC_ST_vec:
        WACC_ST = wacc
        for costGHG in Cost_GHG_perKG_vec:
            Cost_GHG_perKG = costGHG

            Data_materials = Data_materials_SAVE.copy()
            print(f'Running for Scenario = "{scenario}", WACC = {WACC_ST*100:.2f}% and GHG cost = {Cost_GHG_perKG*1000:.2f}$/tonne')
            starttime = time.time()

            if scenario == 'Regular':
                pass
            elif scenario == 'Before without stock':
                conditions = (Data_materials['Delivery before or after job?'].isin(['After', '-> Before']))
                Data_materials.loc[conditions, 'Delivery before or after job?'] = 'Before without stock'
            elif scenario == 'Before with stock':
                conditions = (Data_materials['Delivery before or after job?'].isin(['After', '-> Before']))
                Data_materials.loc[conditions, 'Delivery before or after job?'] = 'Before with stock'
                # Data_materials['Delivery before or after job?'] = np.where(Data_materials['Delivery before or after job?'].isin(['After', '-> B
            else:
                print('Scenario error')

            First_deliveries = Data_materials['Delivery Date'].dropna().min()
            # Determine distance supplier to original delivery
            Data_materials['Original Delivery Date'] = Data_materials['Delivery Date']
            # Data_materials['Original Date difference to job'] = (Data_materials['Done Since'] - Data_materials['Original Delivery Date']).dt.da
            Data_materials['Original Date difference to job'] = Data_materials.apply(lambda row: calc_datediff(row), axis=1)
            Data_materials['Additional stock days'] = (Data_materials['Suggested latest delivery from previous'] - Data_materials['Previous Done
            Data_materials['Original Delivery Port'] = Data_materials['Delivery Port']
            Data_materials['Additional stock days'].fillna((Data_materials['Suggested latest delivery from previous'] - pd.to_datetime(First_deli
            Data_materials['Original Port LAT'] = Data_materials['Original Delivery Port'].map(Port_data.set_index('PortCode')['Port Latitude'])
            Data_materials['Original Port LON'] = Data_materials['Original Delivery Port'].map(Port_data.set_index('PortCode')['Port Longitude'])
            Data_materials['Original Distance Closest Hub to port'] = Data_materials.apply(lambda row: calc_dist_coord_via_hub(row['Closest Hub L
            # Data_materials['Original Transportation Freight Cost V1'] = (Cost_flight * Data_materials['Original Distance Closest Hub to port']
            Data_materials['Original Transportation Freight Cost'] = Data_materials.apply(lambda row: calc_cost(row['Original Distance Closest Hu
            # Data_materials['Original Transportation GHG (grams) V1'] = (GHG_plane_lh * Data_materials['Original Distance Closest Hub to port']
            Data_materials['Original Transportation GHG (grams)'] = Data_materials.apply(lambda row: calc_ghg(row['Original Distance Closest Hub

            Data_materials['Original Transportation GHG Cost'] = Data_materials['Original Transportation GHG (grams)']/1000 * Cost_GHG_perKG

            Data_materials['Original Cost of Capital loss'] = WACC_ST/365.25 * (Data_materials['Average Price'] + Data_materials['Original Transp
            Data_materials['Original Cost of Capital loss'] = Data_materials['Original Cost of Capital loss'].apply(lambda row: max(0,row))
            Data_materials[f'Original Additional Cost of Capital loss (stock)'] = 0

            Data_materials['NO GHG Original Cost of Capital loss'] = WACC_ST/365.25 * (Data_materials['Average Price'] + Data_materials['Original

            Data_materials['NO GHG Original Cost of Capital loss'] = Data_materials['NO GHG Original Cost of Capital loss'].apply(lambda row: max
            Data_materials[f'NO GHG Original Additional Cost of Capital loss (stock)'] = 0

            Data_materials['Original Total Cost'] = Data_materials['Original Transportation Freight Cost'] + Data_materials['Original Transportat

            Data_materials['Original total cost without CoC'] = Data_materials['Original Transportation Freight Cost'] + Data_materials['Original
            Data_materials['NO GHG Original total cost without CoC'] = Data_materials['Original Transportation Freight Cost'] + Data_materials['A



            Avg_spare_price = Data_materials.groupby('Material ID')['Original total cost without CoC'].mean().reset_index()
            Avg_spare_price.rename(columns={'Original total cost without CoC': 'Avg spare price incl original transport'}, inplace=True)

            Avg_spare_price_NO_GHG = Data_materials.groupby('Material ID')['NO GHG Original total cost without CoC'].mean().reset_index()
            Avg_spare_price_NO_GHG.rename(columns={'NO GHG Original total cost without CoC': 'NO GHG Avg spare price incl original transport'}, i

            Data_materials = pd.merge(Data_materials, Avg_spare_price, on='Material ID', how='left')
            Data_materials = pd.merge(Data_materials, Avg_spare_price_NO_GHG, on='Material ID', how='left')



            # Determine foregoing delivery options
            for i in range(1,num_of_prev+1): #Create empty columns
                Data_materials[f'Alternative Date {i}'] = ''
                Data_materials[f'Date difference to (next) job {i}'] = np.nan
                Data_materials[f'Alternative Port {i}'] = ''
                Data_materials[f'Alternative Port LAT {i}'] = np.nan
                Data_materials[f'Alternative Port LON {i}'] = np.nan
                Data_materials[f'Distance Closest Hub to port {i}'] = np.nan
                Data_materials[f'Transportation Freight Cost {i}'] = np.nan
                # Data_materials[f'Transportation GHG {i} (grams) V1'] = np.nan
                Data_materials[f'Transportation GHG {i} (grams)'] = np.nan
                Data_materials[f'Transportation GHG Cost {i}'] = np.nan
                Data_materials[f'Cost of Capital loss {i}']= np.nan
                Data_materials[f'NO GHG Cost of Capital loss {i}']= np.nan
                Data_materials[f'Additional Cost of Capital loss {i} (stock)'] = np.nan
                Data_materials[f'Total Cost {i}'] = np.nan
```

```
        Hist_port_with_del = Hist_port_with_del.sort_values('Arrival Date GMT', ascending=False)
        for index, row in Data_materials.iterrows(): #For each part, fill in the Alternatively defined columns
            # del_date = row['Delivery Date']
            alternative_rows = []
            job_date = row['Done Since']
            sug_del_date = row['Suggested latest delivery from previous']
            max_del_date = row['Max delivery for after jobs']
            if row['Delivery before or after job?'] in ['Before', '-> After', 'Before without stock', 'Before with stock']: #->After means th
                # alternative_rows = Hist_port_with_del[(Hist_port_with_del['Arrival Date GMT']<= sug_del_date)]
                alternative_rows = Hist_port_with_del[(Hist_port_with_del['Arrival Date GMT']<= sug_del_date) & (Hist_port_with_del['Arrival
            elif row['Delivery before or after job?'] in ['After', '-> Before']: #->Before means that it was meant to be delivered after but
                # alternative_rows = Hist_port_with_del[(Hist_port_with_del['Arrival Date GMT']>= job_date) & (Hist_port_with_del['Arrival Da
                alternative_rows = Hist_port_with_del[(Hist_port_with_del['Arrival Date GMT']>= job_date) & (Hist_port_with_del['Arrival Date
                alternative_rows = alternative_rows.sort_values('Arrival Date GMT', ascending=True)
            else:
                alternative_rows = []
                print('Error for finding alternative range')

            if len(alternative_rows) > max_num_alts:
                max_num_alts = len(alternative_rows)


            for i in range(1,num_of_prev+1):
                if i<= len(alternative_rows):
                    # if Alternative_rows.iloc[i-1]['Arrival Date GMT']>= row['Previous Done Since']:
                    Data_materials.at[index,f'Alternative Date {i}'] = alternative_rows.iloc[i-1]['Arrival Date GMT']
                    if row['Delivery before or after job?'] in ['Before', '-> After', 'Before without stock', 'Before with stock']:
                        Data_materials.at[index, f'Date difference to (next) job {i}'] = (row['Suggested latest delivery from previous'] - al
                    elif row['Delivery before or after job?'] in ['After', '-> Before']:
                        Data_materials.at[index, f'Date difference to (next) job {i}'] = (row['Max delivery for after jobs'] - alternative_rc
                    else:
                        pass
                    Data_materials.at[index,f'Alternative Port {i}'] = alternative_rows.iloc[i-1]['PortCode']
                    Data_materials.at[index,f'Alternative Port LAT {i}'] = alternative_rows.iloc[i-1]['Port Latitude']
                    Data_materials.at[index,f'Alternative Port LON {i}'] = alternative_rows.iloc[i-1]['Port Longitude']
                    Data_materials.at[index,f'Distance Closest Hub to port {i}'] = calc_dist_coord(row['Closest Hub Latitude'],row['Closest H
                else:
                    pass

        for i in range(1, num_of_prev+1):
            # Data_materials[f'Distance Supplier to port {i}'] = pd.to_numeric(Data_materials[f'Distance Supplier to port {i}'], errors='coer
            # Data_materials[f'Date difference to (next) job {i}'] = pd.to_numeric(Data_materials[f'Date difference to (next) job {i}'], erro

            # Data_materials[f'Transportation Freight Cost {i}'] = (Cost_flight * Data_materials[f'Distance Closest Hub to port {i}'] + Cost_
            Data_materials[f'Transportation Freight Cost {i}'] = Data_materials.apply(lambda row: calc_cost(row[f'Distance Closest Hub to por
            # Data_materials[f'Transportation GHG {i} (grams) V1'] = (GHG_plane_lh * Data_materials[f'Distance Closest Hub to port {i}'] + GH
            Data_materials[f'Transportation GHG {i} (grams)'] = Data_materials.apply(lambda row: calc_ghg(row[f'Distance Closest Hub to port
            Data_materials[f'Transportation GHG Cost {i}'] = Data_materials[f'Transportation GHG {i} (grams)'] / 1000 * Cost_GHG_perKG
            Data_materials[f'Cost of Capital loss {i}'] = WACC_ST/365.25 * (Data_materials['Average Price'] + Data_materials[f'Transportation
            Data_materials[f'Cost of Capital loss {i}'] = Data_materials[f'Cost of Capital loss {i}'].apply(lambda row: max(0,row))
            # Data_materials[f'Additional Cost of Capital loss {i} (stock)'] = np.where(Data_materials['Delivery before or after job?'] == 'B
            Data_materials[f'Additional Cost of Capital loss {i} (stock)'] = np.where(Data_materials['Delivery before or after job?'] == 'Bef

            Data_materials[f'NO GHG Cost of Capital loss {i}'] = WACC_ST/365.25 * (Data_materials['Average Price'] + Data_materials[f'Transpo
            Data_materials[f'NO GHG Cost of Capital loss {i}'] = Data_materials[f'NO GHG Cost of Capital loss {i}'].apply(lambda row: max(0,r
            # Data_materials[f'Additional Cost of Capital loss {i} (stock)'] = np.where(Data_materials['Delivery before or after job?'] == 'B
            Data_materials[f'NO GHG Additional Cost of Capital loss {i} (stock)'] = np.where(Data_materials['Delivery before or after job?']

            Data_materials[f'Total Cost {i}'] = Data_materials[f'Transportation Freight Cost {i}']  + Data_materials[f'Transportation GHG Cos



        columns = ['Optimal option num', 'Optimal Delivery Date', 'Optimal Date difference to (next) job', 'Optimal Delivery Port', 'Optimal
        columns_original = ['Original option num', 'Original Delivery Date', 'Original Date difference to job', 'Original Delivery Port', 'Or


        optimal_values = Data_materials.apply(find_optimal, axis=1)
        for i, col in enumerate(columns):
            Data_materials[col] = [val[i] for val in optimal_values]

        #### Leave this part out, not picking the optimal situation.
        # cost_condition = Data_materials['Optimal Total Cost'] >= Data_materials['Original Total Cost']
        # Data_materials.loc[cost_condition, 'Optimal option num'] = 'Original'
        # for i in range(1,len(columns)): #This is 1,len because then the optimal num is not considered
        #     Data_materials.loc[cost_condition, columns[i]] = Data_materials.loc[cost_condition, columns_original[i]]

        columns_or = ['Original Transportation Freight Cost', 'Original Transportation GHG (grams)', 'Original Transportation GHG Cost', 'Ori
        columns_opt = ['Optimal Transportation Freight Cost', 'Optimal Transportation GHG (grams)', 'Optimal Transportation GHG Cost', 'Optim
        columns_diff = ['Improvement Transportation Freight Cost', 'Improvement Transportation GHG (grams)', 'Improvement Transportation GHG

        for i in range(0,len(columns_diff)):
            Data_materials[columns_diff[i]] = Data_materials[columns_or[i]] - Data_materials[columns_opt[i]]

        Data_materials['Improvement in tkm'] = Data_materials['Improvement Distance Closest Hub to port'] * Data_materials['Weight']

        globals()[f'Data_materials_save_{scenario}_{wacc}_{costGHG}'] = Data_materials.copy()

        #print results
        globals()[f'Improvement_df_GHGC={costGHG:.2f}$_WACC={WACC_ST:.2f}'] = pd.DataFrame([])
        globals()[f'Improvement_df_GHGC={costGHG:.2f}$_WACC={WACC_ST:.2f}']['Scenario'] = [scenario, scenario, scenario, scenario, scenario,
        globals()[f'Improvement_df_GHGC={costGHG:.2f}$_WACC={WACC_ST:.2f}']['GHG cost per kg'] = [costGHG, costGHG, costGHG, costGHG, costGHG
        globals()[f'Improvement_df_GHGC={costGHG:.2f}$_WACC={WACC_ST:.2f}']['WACC'] = [WACC_ST, WACC_ST, WACC_ST, WACC_ST, WACC_ST,
        globals()[f'Improvement_df_GHGC={costGHG:.2f}$_WACC={WACC_ST:.2f}']['Improvement in'] = ['Freight Cost', 'GHG Cost', 'Cost of Capital
        globals()[f'Improvement_df_GHGC={costGHG:.2f}$_WACC={WACC_ST:.2f}']['Unit'] = ['$','$','$','$','$','tonne CO2-eq','km', 'tkm']


        imp_FC_all = Data_materials['Improvement Transportation Freight Cost'].sum()
        imp_GHGC_all = Data_materials['Improvement Transportation GHG Cost'].sum()
        imp_COC_all = Data_materials['Improvement Cost of Capital loss'].sum()
        imp_ACOC_all = Data_materials['Improvement Additional Cost of Capital loss (stock)'].sum()
        imp_TCOC_all = imp_COC_all+imp_ACOC_all
        imp_COC_all_NOGHG = Data_materials['NO GHG Improvement Cost of Capital loss'].sum()
        imp_ACOC_all_NOGHG = Data_materials['NO GHG Improvement Additional Cost of Capital loss (stock)'].sum()
```

```
        imp_TCOC_all_NOGHG = imp_COC_all_NOGHG+imp_ACOC_all_NOGHG
        imp_all_all = Data_materials['Improvement Total Cost'].sum()
        imp_GHG_all = Data_materials['Improvement Transportation GHG (grams)'].sum()/1000000
        imp_dist_all = Data_materials['Improvement Distance Closest Hub to port'].sum()
        imp_tkm_all = Data_materials['Improvement in tkm'].sum()
        imp_int_cost = imp_FC_all + imp_TCOC_all

        or_FC = Data_materials['Original Transportation Freight Cost'].sum()
        or_GHG = Data_materials['Original Transportation GHG (grams)'].sum()/1000000
        or_GHGC = Data_materials['Original Transportation GHG Cost'].sum()
        or_ACOC = Data_materials['Original Additional Cost of Capital loss (stock)'].sum()
        or_COC = Data_materials['Original Cost of Capital loss'].sum()
        or_TCOC = or_ACOC + or_COC
        or_ACOC_NOGHG = Data_materials['NO GHG Original Additional Cost of Capital loss (stock)'].sum()
        or_COC_NOGHG = Data_materials['NO GHG Original Cost of Capital loss'].sum()
        or_TCOC_NOGHG = or_ACOC_NOGHG + or_COC_NOGHG
        or_total_cost = Data_materials['Original Total Cost'].sum()
        or_int_cost = or_FC + or_TCOC

        opt_FC = Data_materials['Optimal Transportation Freight Cost'].sum()
        opt_GHG = Data_materials['Optimal Transportation GHG (grams)'].sum()/1000000
        opt_GHGC = Data_materials['Optimal Transportation GHG Cost'].sum()
        opt_ACOC = Data_materials['Optimal Additional Cost of Capital loss (stock)'].sum()
        opt_COC = Data_materials['Optimal Cost of Capital loss'].sum()
        opt_TCOC = opt_COC + opt_ACOC
        opt_ACOC_NOGHG = Data_materials['NO GHG Optimal Additional Cost of Capital loss (stock)'].sum()
        opt_COC_NOGHG = Data_materials['NO GHG Optimal Cost of Capital loss'].sum()
        opt_TCOC_NOGHG = opt_COC_NOGHG + opt_ACOC_NOGHG
        opt_total_cost = Data_materials['Optimal Total Cost'].sum()
        opt_int_cost = opt_FC + opt_TCOC

        per_FC = calculate_percentage(imp_FC_all,or_FC)
        per_GHG = calculate_percentage(imp_GHG_all,or_GHG)
        per_GHGC = calculate_percentage(imp_GHGC_all,or_GHGC)
        per_COC = calculate_percentage(imp_COC_all,or_COC)
        per_ACOC = calculate_percentage(imp_ACOC_all,or_ACOC)
        per_TCOC = calculate_percentage(imp_TCOC_all,or_TCOC)
        per_COC_NOGHG = calculate_percentage(imp_COC_all_NOGHG,or_COC_NOGHG)
        per_ACOC_NOGHG = calculate_percentage(imp_ACOC_all_NOGHG,or_ACOC_NOGHG)
        per_TCOC_NOGHG = calculate_percentage(imp_TCOC_all_NOGHG,or_TCOC_NOGHG)
        per_total_cost = calculate_percentage(imp_all_all,or_total_cost)
        per_int_cost = calculate_percentage(imp_int_cost,or_int_cost)


        globals()[f'Improvement_df_GHGC={costGHG:.2f}$_WACC={WACC_ST:.2f}']['Total'] = [imp_FC_all, imp_GHGC_all, imp_COC_all, imp_ACOC_all,

        # Calculate per year
        years = sorted(Data_materials['Year'].unique())
        testcost = 50000.1234
        # Calculate the maximum length of the labels before the equal signs
        max_label_length = max(
            len(f'Improvement in Freight Cost:'),
            len(f'Improvement in GHG Cost:'),
            len(f'Improvement in Cost of Capital loss:'),
            len(f'Improvement in Additional Cost of Capital loss (stock):'),
            len(f'Improvement in Total Cost:'),
            len(f"Improvement in GHG emissions:"),
            len(f"Improvement in distance:"),
            len(f"Improvement in tkm:")
        )
        max_label_length = max_label_length +5

        for year in years:
            DF_year = Data_materials[Data_materials['Year'] == year]
            imp_FC = DF_year['Improvement Transportation Freight Cost'].sum()
            imp_GHGC = DF_year['Improvement Transportation GHG Cost'].sum()
            imp_COC = DF_year['Improvement Cost of Capital loss'].sum()
            imp_ACOC = DF_year['Improvement Additional Cost of Capital loss (stock)'].sum()
            imp_tot = DF_year['Improvement Total Cost'].sum()
            imp_GHG = DF_year['Improvement Transportation GHG (grams)'].sum()/1000000
            imp_dist = DF_year['Improvement Distance Closest Hub to port'].sum()
            imp_tkm = DF_year['Improvement in tkm'].sum()

            globals()[f'Improvement_df_GHGC={costGHG:.2f}$_WACC={WACC_ST:.2f}'][f'{year}'] = [imp_FC, imp_GHGC, imp_COC, imp_ACOC, imp_tot, i
            # print(f'Year: {year}')
            # print(f'{"Improvement in Freight Cost:":<{max_label_length}} ${imp_FC:.2f}')
            # print(f'{"Improvement in GHG Cost:":<{max_label_length}} ${imp_GHGC:.2f}')
            # print(f'{"Improvement in Cost of Capital loss:":<{max_label_length}} ${imp_COC:.2f}')
            # print(f'{"Improvement in Total Cost:":<{max_label_length}} ${imp_tot:.2f}')
            # print(f'{"Improvement in GHG emissions:":<{max_label_length}} {imp_GHG:.2f}[tonne CO2-eq]')
            # print('')

        new_Result_V2 = {
            'Scenario':scenario,
            'GHG cost [$ per kg]':costGHG,
            'WACC':wacc,
            'Reduction in Distance [km]':imp_dist_all,
            'Reduction in tonne-kilometer [tkm]':imp_tkm_all,
            'Reduction in Freight Cost [$]':imp_FC_all,
            'Reduction in GHG Emissions [t]':imp_GHG_all,
            'Reduction in GHG Cost [$]':imp_GHGC_all,
            'Reduction in Cost of Capital [$]':imp_COC_all,
            'Reduction in additional Cost of Capital [$]':imp_ACOC_all,
            'Reduction in Total Cost of Capital [$]':imp_TCOC_all,
            'NO GHG Reduction in Cost of Capital [$]':imp_COC_all_NOGHG,
            'NO GHG Reduction in additional Cost of Capital [$]':imp_ACOC_all_NOGHG,
            'NO GHG Reduction in Total Cost of Capital [$]':imp_TCOC_all_NOGHG,
            'Reduction in Total Cost [$]':imp_all_all,
            'Reduction in Internal Cost [$]':imp_int_cost,
            'Original Freight Cost [$]':or_FC,
            'Original GHG Emissions [t]':or_GHG ,
            'Original GHG Cost [$]':or_GHGC ,
            'Original Cost of Capital [$]':or_COC,
            'Original additional Cost of Capital (stock days) [$]':or_ACOC,
```

```
                'Original Total Cost of Capital [$]':or_TCOC,
                'NO GHG Original Cost of Capital [$]':or_COC_NOGHG,
                'NO GHG Original additional Cost of Capital (stock days) [$]':or_ACOC_NOGHG,
                'NO GHG Original Total Cost of Capital [$]':or_TCOC_NOGHG,
                'Original Total Cost [$]':or_total_cost,
                'Original Internal Cost [$]':or_int_cost,
                'Optimal Freight Cost [$]':opt_FC,
                'Optimal GHG Emissions [t]':opt_GHG ,
                'Optimal GHG Cost [$]':opt_GHGC ,
                'Optimal Cost of Capital [$]':opt_COC,
                'Optimal additional Cost of Capital (stock days) [$]':opt_ACOC,
                'Optimal Total Cost of Capital [$]':opt_TCOC,
                'NO GHG Optimal Cost of Capital [$]':opt_COC_NOGHG,
                'NO GHG Optimal additional Cost of Capital (stock days) [$]':opt_ACOC_NOGHG,
                'NO GHG Optimal Total Cost of Capital [$]':opt_TCOC_NOGHG,
                'Optimal Total Cost [$]':opt_total_cost,
                'Optimal Internal Cost [$]':opt_int_cost,
                'Percentage reduction in Freight Cost':per_FC,
                'Percentage reduction in GHG Emissions':per_GHG,
                'Percentage reduction in GHG Cost':per_GHGC,
                'Percentage reduction in Cost of Capital':per_COC,
                'Percentage reduction in additional Cost of Capital':per_ACOC,
                'Percentage reduction in Total Cost of Capital':per_TCOC,
                'NO GHG Percentage reduction in Cost of Capital':per_COC_NOGHG,
                'NO GHG Percentage reduction in additional Cost of Capital':per_ACOC_NOGHG,
                'NO GHG Percentage reduction in Total Cost of Capital':per_TCOC_NOGHG,
                'Percentage reduction in Total Cost':per_total_cost,
                'Percentage reduction in Internal Cost':per_int_cost
            }

            new_row_df = pd.DataFrame(new_Result_V2, index=[0])
            Results_V2 = pd.concat([Results_V2, new_row_df], ignore_index=True)
            #Save in pickle each loop
            pickle_filename = os.path.join(path, Ship_Name, 'Results_V2.pickle')
            Results_V2.to_pickle(pickle_filename)

            # print(globals()[f'Improvement_df_GHGC={costGHG}$'])
            Results_DF = pd.concat([Results_DF,globals()[f'Improvement_df_GHGC={costGHG:.2f}$_WACC={WACC_ST:.2f}']],ignore_index=True)


########################
########################




            print(f'Time run = {time.time()-starttime} sec')


            # # Combine unique values from both columns
            # all_ports = set(Data_materials["Delivery Port"].unique()) | set(Data_materials["Optimal Delivery Port"].unique())
            #
            # # Calculate value counts for both columns based on the combined unique values
            # value_counts_combined = []
            # value_counts2_combined = []
            #
            # for port in all_ports:
            #     value_counts_combined.append(Data_materials["Delivery Port"].value_counts().get(port, 0))
            #     value_counts2_combined.append(Data_materials["Optimal Delivery Port"].value_counts().get(port, 0))
            #
            # # Sort only the value_counts_combined and keep value_counts2_combined in the same order
            # sorted_indices = np.argsort(value_counts_combined)[::-1]
            # sorted_ports_combined = [list(all_ports)[i] for i in sorted_indices]
            # sorted_value_counts_combined = [value_counts_combined[i] for i in sorted_indices]
            # sorted_value_counts2_combined = [value_counts2_combined[i] for i in sorted_indices]
            #
            # # %% Create the bar plot
            #
            # bar_width = 0.5
            # add_dist = 1.2
            # x_values = range(len(sorted_ports_combined))
            #
            # plt.figure()
            # ax = plt.subplot(1, 1, 1)
            #
            # ax.bar([x * add_dist for x in x_values], sorted_value_counts_combined, width=bar_width, align='center', label='Delivery Port')
            # ax.bar([x * add_dist + bar_width for x in x_values], sorted_value_counts2_combined, width=bar_width, align='center', label='Optimal
            #
            # ax.set_xticks([x * add_dist + bar_width/2 for x in x_values])
            # ax.set_xticklabels(sorted_ports_combined, rotation=45, ha='right')
            #
            # for p in ax.patches:
            #     ax.annotate(str(int(p.get_height())), (p.get_x() + p.get_width() / 2., p.get_height()),
            #                 ha='center', va='center', xytext=(0, 5), textcoords='offset points')
            #
            # plt.xlabel('Delivery Port')
            # plt.ylabel('Number of occurrences')
            # plt.legend()
            # plt.tight_layout()
            #
            # max_label_length = max(len('Scenario'), len('WACC'), len('GHG cost'))
            #
            # title_template = (
            #     'Delivery port occurrences for:' + '\n'
            #     '{:<{width}} = {:<{value_width}}\n'
            #     '{:<{width}} = {:<{value_width}}\n'
            #     '{:<{width}} = {:<{value_width}}'
            # )
            #
            # value_width = max(
            #     len(scenario),
            #     len('{:.2f} %'.format(wacc * 100)),
            #     len('{:.2f} $/tonne'.format(costGHG * 1000))
            # )+5
```

```
            #
            # plt.title(title_template.format(
            #     'Scenario', scenario,
            #     'WACC', '{:.2f} %'.format(wacc * 100),
            #     'GHG cost', '{:.2f} $/tonne'.format(costGHG * 1000),
            #     width=max_label_length,
            #     value_width=value_width
            # ))
            #
            # plot_path = os.path.join(path, Ship_Name, 'Delivery ports plot', f'Results {num_of_prev}alternatives - limited months')
            # os.makedirs(plot_path, exist_ok=True)
            # plt.savefig(os.path.join(plot_path, f"Port_occurences_{scenario}_{wacc*100:.2f}_{costGHG*1000:.2f}.png"), bbox_inches='tight', dpi=
            #
            # plt.show()






#
# Results_V2['Percentage Freight Cost'] = Results_V2.apply(lambda row: calculate_percentage(row['Improvement in Freight Cost'], row['Original Fre
# Results_V2['Percentage GHG emissions'] = Results_V2.apply(lambda row: calculate_percentage(row['Improvement in GHG emissions'], row['Original G
# Results_V2['Percentage GHG Cost'] = Results_V2.apply(lambda row: calculate_percentage(row['Improvement in GHG Cost'], row['Original GHG Cost'])
# Results_V2['Percentage Additional CoC loss'] = Results_V2.apply(lambda row: calculate_percentage(row['Improvement in Additional CoC loss'], row
# Results_V2['Percentage CoC loss'] = Results_V2.apply(lambda row: calculate_percentage(row['Improvement in CoC loss'], row['Original CoC loss'])
# Results_V2['Percentage Total Cost'] = Results_V2.apply(lambda row: calculate_percentage(row['Improvement in Total Cost'], row['Original Total C
#
# Results_V2['Total CoC improvement'] = Results_V2['Improvement in Additional CoC loss'] + Results_V2['Improvement in CoC loss']
# Results_V2['Original Total CoC improvement'] = Results_V2['Original Additional CoC loss'] + Results_V2['Original CoC loss']
# Results_V2['Percentage Total CoC loss'] = Results_V2.apply(lambda row: calculate_percentage(row['Total CoC improvement'], row['Original Total C
Results_V2['Legend'] = Results_V2.apply(lambda row: f"{row['Scenario']} and WACC {int(row['WACC'] * 100)}%", axis=1)
print(f'The maximum number of considered alternatives is: {max_num_alts}')
#%%
# Process results

# unique_legend = Results_V2['Legend'].unique()
# for legend_value in unique_legend:
#     legend_df = Results_V2[Results_V2['Legend']==legend_value]
#
#     plt.plot(legend_df['GHG cost [$ per kg]'], legend_df['Improvement in GHG emissions'], label = legend_value)
#
# plt.xlabel('GHG cost [$ per kg]')
# plt.ylabel('Improvement in GHG emissions [tonne]')
# plt.legend(title='Legend', bbox_to_anchor=(1.04,0.5), loc='center left')
# plt.title('Improvement in GHG emissions')
# plt.show()
#
# for legend_value in unique_legend:
#     legend_df = Results_V2[Results_V2['Legend']==legend_value]
#
#     plt.plot(legend_df['GHG cost [$ per kg]'], legend_df['Improvement in Total Cost'], label = legend_value)
#
# plt.xlabel('GHG cost [$ per kg]')
# plt.ylabel('Improvement in Total Cost [$]')
# plt.legend(title='Legend', bbox_to_anchor=(1.04,0.5), loc='center left')
# plt.title('Improvement in Total Cost')
# plt.show()
#%%
Material_col_names = pd.DataFrame(Data_materials.columns.values)

file_path = os.path.join(path, Ship_Name, "Test sheet with model.xlsx")
sheetname = 'Scenario_WACC_GHGcost'

if os.path.isfile(file_path):
    pass
else:
    wb = openpyxl.Workbook()
    wb.active.title = sheetname
    wb.save(file_path)

with pd.ExcelWriter(file_path, mode="a", engine="openpyxl",if_sheet_exists='replace') as writer:
    Results_DF.to_excel(writer, sheet_name='Results', index=False)
    Results_V2.to_excel(writer, sheet_name='Result V2', index=False)
    Data_materials.to_excel(writer, sheet_name='Data_materials', index=False)
    # for scenario in Scenarios:
    #     if scenario == 'Regular':
    #         sc = 'R'
    #     elif scenario == 'Before with stock':
    #         sc = 'B-S'
    #     elif scenario == 'Before without stock':
    #         sc = 'B-nS'
    #     else:
    #         pass
    #     for wacc in WACC_ST_vec:
    #         for costGHG in Cost_GHG_perKG_vec:
    #             globals()[f'Data_materials_save_{scenario}_{wacc}_{costGHG}'].to_excel(writer, sheet_name=f'{sc}_{wacc}_{costGHG}', index=False


    Material_col_names.to_excel(writer, sheet_name='Columns', index=False)
    Hist_port_with_del.to_excel(writer, sheet_name='Ports with deliveries', index=False)
    Port_data.to_excel(writer, sheet_name='Port_data', index=False)


# Load workbook
wb = openpyxl.load_workbook(file_path)
for ws in wb.worksheets:
    ws.auto_filter.ref = ws.dimensions
wb.save(file_path)
#%%

for Ship_Name in [Ship_Name]:
    plot_data = plot_data_counter
    # %% Boxplot
```

```
    fig, ax = plt.subplots(figsize=(5, 8))
    ax.set_title(f'Deviation to the PMS window T0126 for counter based jobs')
    plot_data.boxplot(column='Interval deviation (negative is before PMS)', ax=ax)
    ax.yaxis.set_major_locator(MultipleLocator(25))
    ax.yaxis.set_major_formatter('{x:.0f}')
    ax.yaxis.set_minor_locator(MultipleLocator(5))
    ax.set_ylim(ymax=230, ymin=-150)
    plt.ylabel('Deviation to the PMS window [%]')



    # Add labels at median, lower whisker, upper whisker, and box edges
    boxplot_data = plot_data['Interval deviation (negative is before PMS)']
    medians = boxplot_data.median()
    lower_quartile = boxplot_data.quantile(0.25)
    upper_quartile = boxplot_data.quantile(0.75)
    iqr = upper_quartile - lower_quartile
    upper_whisker = boxplot_data[boxplot_data<=upper_quartile+1.5*iqr].max()
    lower_whisker = boxplot_data[boxplot_data>=lower_quartile-1.5*iqr].min()
    ax.text(0.9, medians, f'Median: {medians:.1f}%', horizontalalignment='right', verticalalignment='center', fontweight='bold')
    ax.text(0.9, lower_whisker, f'Lower Whisker: {lower_whisker:.1f}%', horizontalalignment='right', verticalalignment='center')
    ax.text(0.9, upper_whisker, f'Upper Whisker: {upper_whisker:.1f}%', horizontalalignment='right', verticalalignment='center')
    ax.text(1.1, lower_quartile, f'Q1: {lower_quartile:.1f}%', horizontalalignment='left', verticalalignment='center')
    ax.text(1.1, upper_quartile, f'Q3: {upper_quartile:.1f}%', horizontalalignment='left', verticalalignment='center')

    # Add missing points:
    # points = boxplot_data[boxplot_data > 200].sort_values().reset_index(drop=True)
    # for index, value in points.items():
    #     ax.arrow(0.6 + index*0.2, (1.85)*100, 0, 20, color='black', width=0.005, head_width=.02, head_length=5)
    #     ax.text(0.6 + index*0.2, (1.83)*100, f'Additional \n point \n @{value:.1f}%', ha='center', va='top')

    plt.tight_layout()
    plt.savefig(os.path.join(path, Ship_Name, 'Deviation PMS plots', f"Boxplot_Deviation_PMS_Percentage_Counter_{Ship_Name}.png"), dpi=400)
    plt.show()

    plot_data = plot_data_month
    # %% Boxplot
    fig, ax = plt.subplots(figsize=(5, 8))
    ax.set_title(f'Deviation to the PMS window T0126 for time based jobs')
    plot_data.boxplot(column='Interval deviation (negative is before PMS)', ax=ax)
    ax.yaxis.set_major_locator(MultipleLocator(25))
    ax.yaxis.set_major_formatter('{x:.0f}')
    ax.yaxis.set_minor_locator(MultipleLocator(5))
    ax.set_ylim(ymax=230, ymin=-150)
    plt.ylabel('Deviation to the PMS window [%]')

    # Add labels at median, lower whisker, upper whisker, and box edges
    boxplot_data = plot_data['Interval deviation (negative is before PMS)']
    medians = boxplot_data.median()
    lower_quartile = boxplot_data.quantile(0.25)
    upper_quartile = boxplot_data.quantile(0.75)
    iqr = upper_quartile - lower_quartile
    upper_whisker = boxplot_data[boxplot_data<=upper_quartile+1.5*iqr].max()
    lower_whisker = boxplot_data[boxplot_data>=lower_quartile-1.5*iqr].min()
    ax.text(0.9, medians, f'Median: {medians:.1f}%', horizontalalignment='right', verticalalignment='center', fontweight='bold')
    ax.text(0.9, lower_whisker, f'Lower Whisker: {lower_whisker:.1f}%', horizontalalignment='right', verticalalignment='center')
    ax.text(0.9, upper_whisker, f'Upper Whisker: {upper_whisker:.1f}%', horizontalalignment='right', verticalalignment='center')
    ax.text(1.1, lower_quartile, f'Q1: {lower_quartile:.1f}%', horizontalalignment='left', verticalalignment='center')
    ax.text(1.1, upper_quartile, f'Q3: {upper_quartile:.1f}%', horizontalalignment='left', verticalalignment='center')

    # Add missing points:
    # points = boxplot_data[boxplot_data > 200].sort_values().reset_index(drop=True)
    # for index, value in points.items():
    #     ax.arrow(0.6 + index*0.2, (1.85)*100, 0, 20, color='black', width=0.005, head_width=.02, head_length=5)
    #     ax.text(0.6 + index*0.2, (1.83)*100, f'Additional \n point \n @{value:.1f}%', ha='center', va='top')

    plt.tight_layout()
    plt.savefig(os.path.join(path, Ship_Name, 'Deviation PMS plots', f"Boxplot_Deviation_PMS_Percentage_Month_{Ship_Name}.png"), dpi=400)
    plt.show()

    plot_data = Data_materials
    # %% Boxplot
    fig, ax = plt.subplots(figsize=(5, 8))
    ax.set_title(f'Deviation to the PMS window T0126 for all jobs')
    plot_data.boxplot(column='Interval deviation (negative is before PMS)', ax=ax)
    ax.yaxis.set_major_locator(MultipleLocator(25))
    ax.yaxis.set_major_formatter('{x:.0f}')
    ax.yaxis.set_minor_locator(MultipleLocator(5))
    ax.set_ylim(ymax=230, ymin=-150)
    plt.ylabel('Deviation to the PMS window [%]')

    # Add labels at median, lower whisker, upper whisker, and box edges
    boxplot_data = plot_data['Interval deviation (negative is before PMS)']
    medians = boxplot_data.median()
    lower_quartile = boxplot_data.quantile(0.25)
    upper_quartile = boxplot_data.quantile(0.75)
    iqr = upper_quartile - lower_quartile
    upper_whisker = boxplot_data[boxplot_data<=upper_quartile+1.5*iqr].max()
    lower_whisker = boxplot_data[boxplot_data>=lower_quartile-1.5*iqr].min()
    ax.text(0.9, medians, f'Median: {medians:.1f}%', horizontalalignment='right', verticalalignment='center', fontweight='bold')
    ax.text(0.9, lower_whisker, f'Lower Whisker: {lower_whisker:.1f}%', horizontalalignment='right', verticalalignment='center')
    ax.text(0.9, upper_whisker, f'Upper Whisker: {upper_whisker:.1f}%', horizontalalignment='right', verticalalignment='center')
    ax.text(1.1, lower_quartile, f'Q1: {lower_quartile:.1f}%', horizontalalignment='left', verticalalignment='center')
    ax.text(1.1, upper_quartile, f'Q3: {upper_quartile:.1f}%', horizontalalignment='left', verticalalignment='center')

    # Add missing points:
    # points = boxplot_data[boxplot_data > 200].sort_values().reset_index(drop=True)
    # for index, value in points.items():
    #     ax.arrow(0.6 + index*0.2, (1.85)*100, 0, 20, color='black', width=0.005, head_width=.02, head_length=5)
    #     ax.text(0.6 + index*0.2, (1.83)*100, f'Additional \n point \n @{value:.1f}%', ha='center', va='top')

    plt.tight_layout()
    plt.savefig(os.path.join(path, Ship_Name, 'Deviation PMS plots', f"Boxplot_Deviation_PMS_Percentage_All_{Ship_Name}.png"), dpi=400)
    plt.show()
```

```python

    plot_data = plot_data_counter
    # %% Boxplot
    fig, ax = plt.subplots(figsize=(5, 8))
    ax.set_title(f'Deviation to the PMS window T0126 for counter based jobs')
    plot_data.boxplot(column='Interval deviation (negative is before PMS) DAYS', ax=ax)
    # ax.yaxis.set_major_locator(MultipleLocator(25))
    # ax.yaxis.set_major_formatter('{x:.0f}')
    # ax.yaxis.set_minor_locator(MultipleLocator(5))
    # ax.set_ylim(ymax=230, ymin=-150)
    plt.ylabel('Deviation to the PMS window [days]')

    # Add labels at median, lower whisker, upper whisker, and box edges
    boxplot_data = plot_data['Interval deviation (negative is before PMS) DAYS']
    medians = boxplot_data.median()
    lower_quartile = boxplot_data.quantile(0.25)
    upper_quartile = boxplot_data.quantile(0.75)
    iqr = upper_quartile - lower_quartile
    upper_whisker = boxplot_data[boxplot_data<=upper_quartile+1.5*iqr].max()
    lower_whisker = boxplot_data[boxplot_data>=lower_quartile-1.5*iqr].min()
    ax.text(0.9, medians, f'Median: {medians:.1f}', horizontalalignment='right', verticalalignment='center', fontweight='bold')
    ax.text(0.9, lower_whisker, f'Lower Whisker: {lower_whisker:.1f}', horizontalalignment='right', verticalalignment='center')
    ax.text(0.9, upper_whisker, f'Upper Whisker: {upper_whisker:.1f}', horizontalalignment='right', verticalalignment='center')
    ax.text(1.1, lower_quartile, f'Q1: {lower_quartile:.1f}', horizontalalignment='left', verticalalignment='center')
    ax.text(1.1, upper_quartile, f'Q3: {upper_quartile:.1f}', horizontalalignment='left', verticalalignment='center')

    # Add missing points:
    # points = boxplot_data[boxplot_data > 200].sort_values().reset_index(drop=True)
    # for index, value in points.items():
    #     ax.arrow(0.6 + index*0.2, (1.85)*100, 0, 20, color='black', width=0.005, head_width=.02, head_length=5)
    #     ax.text(0.6 + index*0.2, (1.83)*100, f'Additional \n point \n @{value:.1f}%', ha='center', va='top')

    plt.tight_layout()
    plt.savefig(os.path.join(path, Ship_Name, 'Deviation PMS plots', f"Boxplot_Deviation_PMS_Percentage_Counter_{Ship_Name}_DAYS.png"), dpi=400)
    plt.show()

    plot_data = plot_data_month
    # %% Boxplot
    fig, ax = plt.subplots(figsize=(5, 8))
    ax.set_title(f'Deviation to the PMS window T0126 for time based jobs')
    plot_data.boxplot(column='Interval deviation (negative is before PMS) DAYS', ax=ax)
    # ax.yaxis.set_major_locator(MultipleLocator(25))
    # ax.yaxis.set_major_formatter('{x:.0f}')
    # ax.yaxis.set_minor_locator(MultipleLocator(5))
    # ax.set_ylim(ymax=230, ymin=-150)
    plt.ylabel('Deviation to the PMS window [days]')

    # Add labels at median, lower whisker, upper whisker, and box edges
    boxplot_data = plot_data['Interval deviation (negative is before PMS) DAYS']
    medians = boxplot_data.median()
    lower_quartile = boxplot_data.quantile(0.25)
    upper_quartile = boxplot_data.quantile(0.75)
    iqr = upper_quartile - lower_quartile
    upper_whisker = boxplot_data[boxplot_data<=upper_quartile+1.5*iqr].max()
    lower_whisker = boxplot_data[boxplot_data>=lower_quartile-1.5*iqr].min()
    ax.text(0.9, medians, f'Median: {medians:.1f}', horizontalalignment='right', verticalalignment='center', fontweight='bold')
    ax.text(0.9, lower_whisker, f'Lower Whisker: {lower_whisker:.1f}', horizontalalignment='right', verticalalignment='center')
    ax.text(0.9, upper_whisker, f'Upper Whisker: {upper_whisker:.1f}', horizontalalignment='right', verticalalignment='center')
    ax.text(1.1, lower_quartile, f'Q1: {lower_quartile:.1f}', horizontalalignment='left', verticalalignment='center')
    ax.text(1.1, upper_quartile, f'Q3: {upper_quartile:.1f}', horizontalalignment='left', verticalalignment='center')

    # Add missing points:
    # points = boxplot_data[boxplot_data > 200].sort_values().reset_index(drop=True)
    # for index, value in points.items():
    #     ax.arrow(0.6 + index*0.2, (1.85)*100, 0, 20, color='black', width=0.005, head_width=.02, head_length=5)
    #     ax.text(0.6 + index*0.2, (1.83)*100, f'Additional \n point \n @{value:.1f}%', ha='center', va='top')

    plt.tight_layout()
    plt.savefig(os.path.join(path, Ship_Name, 'Deviation PMS plots', f"Boxplot_Deviation_PMS_Percentage_Month_{Ship_Name}_DAYS.png"), dpi=400)
    plt.show()

    plot_data = Data_materials
    # %% Boxplot
    fig, ax = plt.subplots(figsize=(5, 8))
    ax.set_title(f'Deviation to the PMS window T0126 for all jobs')
    plot_data.boxplot(column='Interval deviation (negative is before PMS) DAYS', ax=ax)
    # ax.yaxis.set_major_locator(MultipleLocator(25))
    # ax.yaxis.set_major_formatter('{x:.0f}')
    # ax.yaxis.set_minor_locator(MultipleLocator(5))
    # ax.set_ylim(ymax=230, ymin=-150)
    plt.ylabel('Deviation to the PMS window [days]')

    # Add labels at median, lower whisker, upper whisker, and box edges
    boxplot_data = plot_data['Interval deviation (negative is before PMS) DAYS']
    medians = boxplot_data.median()
    lower_quartile = boxplot_data.quantile(0.25)
    upper_quartile = boxplot_data.quantile(0.75)
    iqr = upper_quartile - lower_quartile
    upper_whisker = boxplot_data[boxplot_data<=upper_quartile+1.5*iqr].max()
    lower_whisker = boxplot_data[boxplot_data>=lower_quartile-1.5*iqr].min()
    ax.text(0.9, medians, f'Median: {medians:.1f}', horizontalalignment='right', verticalalignment='center', fontweight='bold')
    ax.text(0.9, lower_whisker, f'Lower Whisker: {lower_whisker:.1f}', horizontalalignment='right', verticalalignment='center')
    ax.text(0.9, upper_whisker, f'Upper Whisker: {upper_whisker:.1f}', horizontalalignment='right', verticalalignment='center')
    ax.text(1.1, lower_quartile, f'Q1: {lower_quartile:.1f}', horizontalalignment='left', verticalalignment='center')
    ax.text(1.1, upper_quartile, f'Q3: {upper_quartile:.1f}', horizontalalignment='left', verticalalignment='center')

    # Add missing points:
    # points = boxplot_data[boxplot_data > 200].sort_values().reset_index(drop=True)
    # for index, value in points.items():
    #     ax.arrow(0.6 + index*0.2, (1.85)*100, 0, 20, color='black', width=0.005, head_width=.02, head_length=5)
    #     ax.text(0.6 + index*0.2, (1.83)*100, f'Additional \n point \n @{value:.1f}%', ha='center', va='top')

    plt.tight_layout()
    plt.savefig(os.path.join(path, Ship_Name, 'Deviation PMS plots', f"Boxplot_Deviation_PMS_Percentage_All_{Ship_Name}_DAYS.png"), dpi=400)
```

```
    plt.show()
#%%

#%%

```