Empty Container Logistics Optimization: An Implementation Framework and Methods

by

Bin Hong Alex Lee

Bachelor of Engineering (Honors) in Computer Engineering
National University of Singapore (2006)

Submitted to the System Design and Management Program
in partial fulfillment of the requirements for the Degree of
Master of Science in Engineering and Management

at

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Cambridge, Massachusetts

[February 2014]
Jan 17, 2014

© 2014 Bin Hong Alex Lee. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Signature redacted

Signature of Author: 

Signature redacted

Certified by: __________________________

Chris Caplice, Thesis Supervisor
Sr Lecturer, Engineering and System Division &
Executive Director, Center for Transportation and Logistics

Signature redacted

Accepted by: __________________________

Patrick Hale
Director, System Design and Management Fellows Program
This page is intentionally left blank
Empty Container Logistics Optimization: An Implementation Framework and Methods

by

Bin Hong Alex Lee

Submitted to the System Design and Management Program in partial fulfillment of the requirements for the Degree of

Master of Science in Engineering and Management

ABSTRACT

Empty container logistics is a huge cost component in an ocean carrier’s operations. Managing this cost is important to ensure profitability of the business. This thesis proposes a 3-stage framework to handle empty container logistics with cost management as the objective. The first stage studies the forecasting of laden shipment demand, which provides the empty container supply requirement. Based on the supply needs, the problem of optimizing the fleet size was then addressed by using an inventory model to establish the optimal safety stock level. Simulations were used to understand the sensitivity of safety stock to desired service level. The final stage involves using mathematical programming to optimize repositioning costs incurred by carriers to ship empty containers to places which need them due to trade imbalance. At the same time, costs that are incurred due to leasing and storage are considered. A comparison between just-in-time and pre-emptive replenishment was performed and impact due to uncertainties is investigated. The framework is then implemented in a Decision Support System for an actual ocean carrier and is used to assist the empty container logistics team to take the best course of action in daily operations. The results from the optimizations show that there are opportunities for the carrier to reduce its fleet size and cut empty container logistics related costs.
Acknowledgement

I dedicate this thesis to my family and friends for their support in my further studies with special mention to Sheryl F. for her moral support during the writing of this thesis and helping to proofread. I also wish to thank my colleagues in Singapore, Australia and China for taking time off their busy schedule to help me obtain and understand the data used in this thesis. In addition, I like to thank my mentor, Eng Soon T., for providing his advice on subject matter, Joshua L. for providing his SAP APO expertise, Kristine J. for introducing me to R which I used in my implementation of the Decision Support System, and Hanzhong C. for providing feedback.

Lastly, I am very grateful to Prof Chris Caplice for offering his invaluable guidance and insights, without which, this thesis would not have been possible.
Contents

Chapter 1 ........................................................................................................................................................................... 1
  1.1 Background ......................................................................................................................................................... 1
  1.2 Motivation .......................................................................................................................................................... 2

Chapter 2 ........................................................................................................................................................................... 4
  2.1 Empty Container Logistics ............................................................................................................................... 4
  2.2 Empty Container Logistics Optimization Framework .................................................................................. 5

Chapter 3 ......................................................................................................................................................................... 8
  3.1 Demand Management ........................................................................................................................................ 8
  3.2 Demand Planning and Forecasting .................................................................................................................. 8
  3.3 Assessing Forecast Strategy ............................................................................................................................ 10
  3.4 Demand Plan Review ......................................................................................................................................... 13
  3.5 Demand Distribution .......................................................................................................................................... 16
  3.6 Demand Management Case Studies ............................................................................................................. 17
    3.6.1 Top-down Time Series Analysis ................................................................................................................. 19
    3.6.2 Bottom-Up Time Series Analysis .............................................................................................................. 24

Chapter 4 ....................................................................................................................................................................... 25
  4.1 Customer Service Level and Safety Stock ................................................................................................. 25
  4.2 Multi-period Inventory Model ........................................................................................................................ 26
  4.3 Inventory Model and Safety Stock Calculation ........................................................................................... 28
  4.4 Service Level Monte Carlo Simulation ......................................................................................................... 30

Chapter 5 ....................................................................................................................................................................... 32
  5.1 Empty Container Repositioning ...................................................................................................................... 32
  5.2 Leasing and Holding Empty Container Inventory ....................................................................................... 35
  5.3 Problem Formulation ....................................................................................................................................... 36
  5.4 Just-in-Time Optimization Model .................................................................................................................. 40
  5.5 Just-in-Time Optimization Results ................................................................................................................. 43
  5.6 Pre-emptive Repositioning and Inventory Holding ..................................................................................... 46
5.7 Pre-emptive Reposition Optimization Model ................................................................. 50
5.8 Pre-emptive Optimization Results .................................................................................. 51
5.9 Improving Optimization's Computation time ................................................................. 56
5.10 Uncertainty Consideration and Robust Optimization .................................................. 58

Chapter 6 ....................................................................................................................................................................... 61
6.1 Business Analytics and System Implementation ......................................................... 61
6.2 Policy and Impact Discussion ....................................................................................... 64
6.3 Conclusion .................................................................................................................................................. 66
6.4 Future Work .............................................................................................................................................. 67

Bibliography ................................................................................................................................................................. 68

Appendix A - Forecasting Source Code in R ........................................................................ 71
Appendix B - Empty Logistics Optimizer Gurobi Source Code ......................................... 74
List of Figures and Tables

Figure 1 Container Logistics Flow .................................................................................................................... 4
Figure 2 - Empty Containers Logistics Optimization Framework ............................................................................... 7
Figure 3 – Dynamics of Container Shipping Logistics ......................................................................................... 14
Figure 4 - Export and Import return volume of Port AU ....................................................................................... 18
Figure 5 - Export and Import return volume of Port CN ....................................................................................... 18
Figure 6 - Decomposition of Export and Import Return for Port AU and CN over the years .......20
Figure 7 – Forecast vs Actual for Port AU and CN ............................................................................................... 21
Figure 8 – Autocorrelation check for forecast errors ............................................................................................. 22
Figure 9 – Residual checks for forecast errors ........................................................................................................ 22
Figure 10 – Numerical results for testing Forecast Errors ....................................................................................... 23
Figure 11 - Multi-period Forecasting and Inventory Model ..................................................................................... 26
Figure 12 – Network Flow w.r.t. a Port P₁ ................................................................................................................ 36
Figure 13 – Scenarios for Surplus and Deficit Ports .............................................................................................. 37
Figure 14 – Inbound Distribution Plans for Port AU and CN .................................................................................. 45
Figure 15 – Consumption Window of Empty Containers ....................................................................................... 45
Figure 16 – Container Network Flow for Pre-emptive Repositioning .................................................................. 47
Figure 17 – Container Network Flow Space-Time Diagram ..................................................................................... 48
Figure 18 – Empty Logistics Cost against Inventory Holding Policy ..................................................................... 52
Figure 19 – Stock Level against Inventory Holding Policy ....................................................................................... 52
Figure 20 – Cost Breakdown by Empty Containers Supplier Country for Port AU ............................................. 53
Figure 21 – Cost Breakdown by Empty Containers Supplier Country for Port CN .............................................. 54
Figure 22 – Inbound Distribution Plans with Pre-emptive Repositioning for Port AU and CN 55
Figure 23 – Computation Time against Inventory Holding Policy ........................................................................... 56
Figure 24 – Problem Reduction ............................................................................................................................. 57
Chapter 1

1.1 Background

Containerization increases the efficiency of loading, unloading and stacking cargo on multiple freight transportation modes such as ships, rail and trucks. The introduction of standard size containers in the 1950s was an important factor in improving the efficiency. These standard size containers primarily consist of 20, 40, 45 and 53-footer General Purpose, Reefer or Special Type containers. The efficient stowing and lashing of the containers enable reliable transportation over long distances and transfer from one mode to another. In fact, the use of containers has now become ubiquitous in international commodity transportation and has greatly contributed to globalization as a whole.

In the last few decades, the world has seen a huge growth of laden containers transported to and from countries and ports. The flow of containers to different regions is, however, not balanced. Depending on the nature of industry and resources which require cargo shipments, a country or port may be trade surplus or deficit, as measured by export against import volume. This means that trade surplus areas face a shortage of empty containers for shipments while trade deficit areas see an accumulation of empty containers in the depots. This imbalance resulted in the biggest hidden costs to the shipping industry with carriers having to carry out backhaul of empty containers. According to Shipping watch (Kristiansen, 2012), the largest ocean carrier in the world, Maersk Line, declared in 2012 that 4 million empty containers were transported on their vessels and up to USD 1 billion was spent in shipping empty containers from surplus to deficit locations. The imbalance in trade between Europe and Asia is one major reason which contributed to this phenomenon.
For the past few years, there was a slump in worldwide container freight rates and up to 20% of major container carriers are suffering losses due to overcapacity and low margins. The low freight rates make it tough for companies to cover the industry's high fixed costs and operating expenses. It became imperative for businesses to optimize costs, especially in empty container logistics, to maintain profitability. Carriers' empty container logistics departments, who are responsible for planning and meeting the shortage requirements with empty container repositioning or short-term leasing of containers, are now more cost conscious. The decisions that they make daily on empty containers logistics affect a significant portion of operating expenses. These expenses include empty container repositioning costs which mainly consist of stevedore, pickup handling, trucking and feeder costs. Leasing of empty containers incurs lift-on lift-off handling costs and higher leasing per diem costs for short term leases. In addition, empty containers, stowed in depots or container yards, also incur storage costs. All these mentioned costs have to be contained in the cost management efforts of the empty container logistics departments.

1.2 Motivation

This thesis proposes a framework and methods to handle empty container logistics efficiently in an ocean carrier. The primary objectives of the framework are to:

1) Introduce systematic processes to model the dynamics of container supply and demand for better management

2) Optimize the carrier's container fleet and cost for empty container replenishment through repositioning (also known as backhaul) and leasing
Through the proposed framework and methods, the questions that this thesis seeks to answer are as below:

1) What is the optimum container stock to keep at different locations (considering trade surplus, trade neutral and trade deficit locations)?

2) What is the strategy for optimal empty container repositioning for replenishment for the trade deficit locations?

By answering the questions above, the carrier will be able to optimize empty container storage cost (for inventory holding) and empty container repositioning and leasing cost (for replenishment). The proposed framework and strategies can be implemented in a carrier's global Enterprise Resource Planning and Decision Support System which can assist the decision makers in different parts of the world to take consistent, accurate and timely actions. In addition, by following the process prescribed in the framework, a better estimation of costs can be obtained for cash flow planning and budgeting in the company.
Chapter 2

2.1 Empty Container Logistics

To develop an end-to-end optimization strategy, we first have to understand the container logistics workflow. The figure below illustrates a typical empty and laden container flow to and from the carrier's and customer's premises and also in transit.

Figure 1. Container Logistics Flow

Containers are assets of carriers and are stored at the carriers' designated depots. In laden container logistics, after a booking confirmation, an empty container is picked up by a shipper...
at the place of origin or arranged to be transported from the depot to the shipper's premise for cargo stuffing. After the container is stuffed, the laden container will be trucked to the port of loading terminal to be loaded on board the carrier's ocean vessel. Upon completion of the ocean transit (which may involve transshipment at different ports), the container is unloaded at the port of discharge. The cargo consignee will pick up or arrange to transport the laden container to its premises (or called place of destination) for unstuffing. Sometimes, the stuffing and unstuffing may take place in designated container freight stations. After cargo is devanned from the container, the empty container is returned to the nearest carrier's depot or designated location by trucking, which can then be used in another shipment booking. In the event where there is a need to replenish empty container's supply from another location, the empty logistics process will proceed. The trucking company will truck the empty container from the origin location to the port upon receipt of work orders from the ocean carrier. The empty container will then be loaded onto the ocean vessel and shipped to the destination location. At the destination location, the empty container may be picked up directly by the shipper from the port or the container may be trucked to the carrier's assigned depot or location.

2.2 Empty Container Logistics Optimization Framework

Optimization of empty container logistics can be performed on two areas, namely cost and total container fleet size. The process involves tight collaboration among the offices in different locations to provide reliable reviews of demand forecasts and take actions on empty containers replenishment work orders. The optimization also requires a lot of information such as historical data and inputs from the departments in different locations. As such, a
highly integrated Enterprise Resource Planning (ERP) platform has to be in place to provide a single source of data and access for the different locations.

Before designing the platform, the introduction of this framework and its methods herein is to first propose standardized systematic processes for ocean carriers to handle empty container logistics across different locations. It starts with modelling the container supply and demand for better management. This is followed by the optimization of the carrier's container fleet and cost for empty container replenishment through repositioning and leasing. The primary goal of the optimization framework is to optimize the cost and total container fleet size.

Figure 2 illustrates the framework. As shown, the framework is divided into three different stages:

1) Demand Management
2) Supply Network Planning
3) Supply Chain Execution (Detailed scheduling)

Each stage has its own processes and the processes are performed by different departments in the ocean carrier. The rest of this thesis will describe these individual processes in detail. Chapter 3 will describe the processes of demand forecasting and demand plan review in the *Demand Management* stage. These processes are carried out by the laden containers demand planning team. Chapter 4 will describe the supply requirement process in the *Supply Network Planning* stage. These processes are to be carried out by the empty container logistics team. Chapter 5 will explore the optimization process with mathematical programming and understand how inventory policies and uncertainties will affect the empty container logistics cost and inventory level. The outputs from this process are an optimized distribution plan for empty containers to meet customers' demands and an optimized container fleet guideline.
Chapter 6 describes the implementation of the framework in a decision support system and discusses the insights obtained from the optimization framework and methods.

Figure 2 - Empty Containers Logistics Optimization Framework
Chapter 3

3.1 Demand Management

Demand management requires a holistic view of the business and is more than just forecasting alone. It consists of activities which are associated with understanding the industry and enterprise outlook, discovering markets and their sentiments, planning products and services for the markets and striving to fulfill customers' demands. This demand planning process often involves not only collaboration between offices in an enterprise but also with the trade partner network. In the case of empty container logistics, demand management information will provide the export and import demand for each port which can then be used to determine the net demand requirement for empty containers. The next few sections will discuss demand planning, forecasting and reviewing in detail.

3.2 Demand Planning and Forecasting

Demand forecasting is making an estimate of what will happen to demand volume in the future. At this stage in the supply chain, analysts are primarily concerned with obtaining estimates of unconstrained demand. Unconstrained demand is the volume required by the business or customers without the constraints of capacity and availability on the provider. The information obtained is useful for downstream operations and inventory planning to maximize profitability.

The forecast procedure in demand planning can be done both qualitatively and quantitatively. Quantitative calculations to obtain estimates are often practiced in companies and many of
these techniques involve statistical methods. There are many reasons why we need the estimates from forecasting. From the perspective of different departments in a company, the uses of forecast in a typical shipping company’s departments are as follows:

*Marketing* – The forecasted demand allows marketing to segment the market and channel their resources appropriately

*Sales* – Forecast information set by marketing can be a benchmark to understand what volumes are expected in each period and evaluate the sales pipeline to support the benchmark

*Inventory* – Demand forecasts will indicate the number of containers that are required in the forecast period so that actions to provide empty containers through empty returns, repositioning and lease hire can be prescribed

*Vessel operations* – The forecasted information can be used to develop and evaluate capacity needs and perform appropriate slot allocations

*Finance* – Demand forecast can be used for calculating budget and accruing expenses. For example, accrual cost can be calculated for the estimated volume of empty container repositioning and leasing to satisfy the laden shipment

To make a calculated estimate of what will happen to the demand volume in the future, information can be derived from past history and using it to extrapolate to the future. The calculation may also have present variables which may affect the extrapolation. These variables include planned activities like freight hike, discount or foreseeable exogenous conditions like competitors’ entry and reaction and market condition etc. (Silver, Pyke, & Rein 1998). It is also noted that companies are well advised to combine statistical analysis with
managerial judgment about short, medium, and long-term outlooks for company sales to customers and markets (Shapiro, 2007). In summary, there are two components of performing forecasting in Demand Planning (Simchi-Levi, Kaminsky, & Simchi-Levi, 2008):

1) **Demand Forecast**: A process in which historical demand data is used as a basis to develop short or long timer estimates of future demands. Usually, quantitative methods are used in this process which may take into account patterns of trend and seasonal effects.

2) **Demand Shaping**: A process which makes use of the present variables mentioned above (outlooks of freight hike, discount and other exogenous conditions) and determine the impact on the demand forecasts (increasing or decreasing the demand)

It is highlighted that since no forecast is ever completely accurate, the results for demand forecast and demand shaping have to indicate the forecasted value and its accuracy (Simchi-Levi et al., 2008). The accuracy of the forecast (or forecast error) can be measured using *Root Mean Square Error*. High demand forecast error has a detrimental effect on supply chain performance, and in our case, will cause inefficient usage of the containers for shipments and incur costs for storage and order for repositioning. We will see the effect forecast errors have on safety stock in Chapter 4.

### 3.3 Assessing Forecast Strategy

Demand forecasting models usually involves projecting future demand patterns based on historical data of past demand. Time series analysis is an important approach for developing forecasts from historical data. Typically, managers use regression (linear regression with time)
and forecast smoothing techniques (like moving average and Holt-Winter and Brown's exponential smoothing) to estimate average demand and demand variability over time. Patterns in historical data such as seasonality or trend that produce a good fit to the data are often investigated and considered. Analysis of trend and seasonal time series data are performed by Holt-Winter method. More complicated procedures by Lewandowski (1982) are also introduced to project forecast for horizon of more than seven periods.

A typical time series analysis can be decomposed into five different elements which are 1) Level, 2) Trend, 3) Seasonality, 4) Cyclicity and 5) Irregularity (Shmueli, Patel, & Bruce, 2010). Level is the deseasonalized demand. Trend is a long term behavior (which is often straight line increase of exponential growth). Seasonality shows the repeating effects of time-of the year. Cyclicity reflects the gradual ups and downs and is different from repeating each year. Irregularity is short term, random and non-systematic noise which we cannot model. There are basically two extrapolation methods for handling seasonality. The most frequent way is to de-seasonalize the data before performing forecasting and then readjust the forecast for seasonality. Seasonality models are usually classified as additive or multiplicative. An additive model finds the seasonal indexes for each month which are then added to the de-seasonalized mean for adjustment of a month's value. A multiplicative model finds the seasonal indexes for each month too, but these indexes are multiplied with the deseasonalized mean instead. Multiplicative models are typically used for seasonality models because they are easier for interpretation.

Lapide (2006) described the benefits of strategy of using top-down and bottom-up forecasting. Top-down forecasting can improve the accuracy of detailed forecast as aggregated information is less volatile than its individual components. Random variations and
errors in individual components will cancel out each other. However, it is noted that top-down forecasting only makes sense if a top-level aggregated group is made up of components with similar patterns of variation when it is used to break down using proportions to the total. The paper also described bottom-up forecasting as better for situations where individual components have different patterns of variations. In this case, the individual forecasts are added up to form the aggregated forecast. The paper then proposed using a combination of top-down and bottom-up forecasting which it described as useful in Sales and Operations Planning process where the development of an accurate baseline forecast can be achieved in using both in conjunction.

In summary, we define the *Demand Forecast Process* as below:

<table>
<thead>
<tr>
<th>Demand Forecast Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Roles and responsibilities of the organization on laden export and import demand, including Trade (Product Development), Marketing, Customer Service and Corporate Development, to be clearly identified</td>
</tr>
<tr>
<td>2) Retrieve historical data from Enterprise Resource Planning system and choose appropriate demand forecast models</td>
</tr>
<tr>
<td>3) Assess the demand forecast needs of the down-stream customer using Top-down and Bottom-up strategy</td>
</tr>
<tr>
<td>4) Sales Forecasting process to include promotions, seasonal planning and collaboration with Customer Service and CRM process in the organization</td>
</tr>
</tbody>
</table>
3.4 Demand Plan Review

After the demand forecast is created, a demand plan review process has to be carried out. This process is a collaborative process and it is indicated in our framework as *Collaborative Demand Plan Review*. In this process, the following steps are to be taken.

**Demand Plan Review Process**

1. Go through consensus process for demand forecast using forecast review reports
2. Reconcile forecast and validate final forecast consensus
3. Check exception management for demand and supply volatility using data mining (weather, commodity demand etc)
4. Provide constant feedback and take action to improve forecast accuracy
5. Define Value Chain Metrics including key measurements for Demand Chain (ie. Forecast vs Actual accuracy, Customer Service Level) to measure forecast accuracy and determine the performance of the entire demand chain
6. Identify empty repositioning constraints and reverse feedback.

We need to understand the system to review the demand plan for better accuracy. A good way to do so is to study the dynamics of empty container logistics in a system dynamics model which is shown in Figure 3. The model's objective is to understand the cost component (Inventory Cost) and how it is affected by exogenous factors and management levers. Exogenous conditions include those highlighted in orange: 1) Port Export and Import.
Demand, 2) Customer detention period (holding of empty containers after cargo devanning) and 3) Empty Repositioning delay (port dwell and ocean transit).

In the model, we also identified several management levers which are highlighted in green, that can affect the objective of minimizing empty logistics cost. They include:

1) Desired Safety Stock
2) Liner Empty Repositioning and Leasing Ratio
3) Liner Waiver Approval Criteria
4) Liner Free Days
5) Liner Export Discount
*Desired Safety Stock* is the level of buffer stock that is required to satisfy uncertain demands. Increase in desired safety stock to meet customers' demands will cause an increase in inventory level gap or shortfall. This will increase the empty repositioning order and container leasing, both of which incur cost. *Liner Empty Repositioning and Leasing Ratio* is the owned container percentage a carrier maintains of its fleet. A carrier can choose to lease containers from container leasing companies to meet customers' demand. Usually, short term leasing cost is much higher than the ownership cost and leasing is utilized mainly only to meet short term demands.

To encourage customers to return containers in the shortest time possible, carriers start to charge consignees detention charges after a period of free days denoted by *Liner Free Days*. Shorter free-days period will lead to higher return rate of empty containers. This not only shortens the turnaround time of the containers but also lowers the total container fleet size requirement. *Liner Waiver Approval Criteria* is a waiver of a percentage of the detention charges incurred by the customers or extension of the free days for important customers. The waivers are approved on a case by case basis and have to be handled with diligence and care as the waivers have the same effect on the return rate of empty containers like the free days.

Due to trade imbalance among different locations, empty repositioning has to be carried out to offset the imbalance. Alternatively, carriers can try to offset part of the imbalance by encouraging export from empty container surplus locations to deficit locations by offering a discount on export denoted by *Liner Export Discount*. The revenue gained on these exports to move the containers from surplus to deficit areas will help to recover the cost of empty
repositioning which otherwise has to be performed. Some carriers may give up to 20% discount on freight to move these containers out of deficit locations.

Other minor factors include 1) Liner container offhire rate (containers which are decommissioned due to state or lack of demand and 2) Maintenance and repair (containers which are out of service due to repair or upgrade requirements). It is important to note that management levers 3-5) are subjected to market forces and responses. As such, for this thesis, we will focus mainly on 1) Desired Safety Stock and 2) Liner Empty Repositioning and Leasing Ratio in our 2-stage optimization.

3.5 Demand Distribution

Finding an accurate model for the demand distribution is an ongoing research effort. Some existing approaches assume potentially oversimplified distributions to create a tractable lead time distribution. It is however important to have an accurate distribution model to determine correct safety stock inventory and reorder point which this research is trying to achieve.

Cobb (2012) described a mixture distribution approach to modeling demand in a continuous review inventory system. The mixture distribution technique that was described employs a mixture of truncated exponentials approximation to the input distributions. This allowed the distributions for lead time and demand per unit time to follow any standard or empirical probability function. The technique is then illustrated by applying to a “normal-gamma” inventory problem, and then by modeling a problem with empirical distributions for lead time and demand per unit time. Since demand is accumulated over time, transit time
distributions will also impact the demand requirement at vessel arrival. Das (2013) found that transit times are usually skewed normal distributions and often displays bimodality. She examines the effects of transit time variability on demand over transit time and the associated costs. Our model, however, shows that using a normal distribution assumption suffices for safety stock estimates.

3.6 Demand Management Case Studies

We looked at the export and import patterns of one of the Australian ports and China ports of a carrier (an Alphaliner Top 100 carrier with a network of 197 ports) and analyzed the export and import demands of two of the ports in the network.

In the Australian port’s data (denoted by Port AU), high volume of export is observed in March, July and August for export out of the Australian ports. They are largely contributed by the few high volume exporters of the company that are mining companies. March is normally the peak season because the wet season usually ends in February and during the wet season, mining does not operate in full mode. There are usually two peaks in July and Aug and November where operations and production will ramp up especially in November as it is before the end of the year holidays and before the wet season kicks in. Information like this can be obtained from the customers and their forecast can be incorporated in the Demand Plan Review Process introduced in Section 3.4 to adjust the carrier’s forecast. There was however some deviation to expected trends in year 2013, where the export saw a jump in the volume as a result of a new mining exporter on-board. Smelting issues at one of the mining companies also contributed to an increase in demand for containerization.
The Chinese port (denoted by Port CN) displays trends where volume is affected by the festive seasons. The Chinese port is a main exporter of fertilizer, chemical, tyres and vegetables and fruits. There are several significant peaks and troughs in the export cycle of the year, some of which are contributed by harvest seasons. However, the main peaks in volume occur in the last two months of the year before the most important Chinese festive period. Shippers are mostly rushing to complete their exports before the Lunar Chinese New Year. There is then significantly lower volume in the month of Lunar Chinese New Year which takes place either in January or February. The volume starts to pick up after the festive season.
For Port CN, there was also a spike in the export volume in the second quarter of year 2013 due to the events where partner carriers' volume spilled over because of problems with their vessels. The increase is seen as temporary.

### 3.6.1 Top-down Time Series Analysis

We observe from the data in Figure 4 and Figure 5, even with irregularities, that there are seasonality trends especially in the export volume. As for trend, it seems that there is slight increase in a year-on-year volume for both ports. To investigate further, we made use of a popular statistical tool, R (R Development Core, 2008), to decompose the data as below. The source code for R can be found in Appendix A.

The time analysis decomposition gives us a clear picture of the deseasonalized volume (level), the change i.e. increase or decrease (slope) and seasonal pattern (season). The data indicates that there is a significant upwards trend for the export and import volume for Port AU, while
there is little increase for the export volume for Port CN. The import volume for Port CN is decreasing year-on-year. For both ports, there are clear seasonal patterns identified by the decomposition.

![Port CN Export Data](image1)

![Port CN Import Data](image2)

**Figure 6 - Decomposition of Export and Import Return for Port AU and CN over the years**

We used actual shipping volume data of Port AU and CN and partitioned the sets into training and validation data sets. For our forecasting, we use Holt Winter seasonal forecast extrapolation. The function used in R was `ets` function (exponential smoothing state space model in R) with multiplicative errors, additive trend and multiplicative seasonality. Multiplicative errors and seasonality are typically used because they are easier for interpretation. Data from 2009 to 2011 is used as training sets and data from 2012 and 2013 are used as validation sets. In our forecast process, we focus on the export volume of the ports as the import volume can be calculated stochastically based on the breakdown of export volume from other ports in the network.
The above figure shows the demand export forecast for the two ports that we studied. We observed huge spikes in export demand for Port CN in the month of June and July because of spill over from partner carriers who had problems with their vessels and load. These increases were temporary and were considered special projects. Therefore, the forecast for Port CN shows that using historical extrapolation does not work all the time as it did not capture the uncertainty. However, the variability and uncertainty of demand can be mitigated in the Demand Plan Review process which we have discussed in Chapter 3.4. Below are the results of error rates of the training and validation set. The training Mean Absolute Percent Error (MAPE) are 21.7% and 17.8% for Port AU and CN respectively. The MAPE for validation sets are 15.1% and 14.5% respectively.

<table>
<thead>
<tr>
<th>Port AU</th>
<th></th>
<th>Port CN</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Set</td>
<td>RMSE</td>
<td>MAPE</td>
<td>Data Set</td>
</tr>
<tr>
<td>Training</td>
<td>169.4</td>
<td>21.7%</td>
<td>Training</td>
</tr>
<tr>
<td>Validation</td>
<td>158.1</td>
<td>15.1%</td>
<td>Validation</td>
</tr>
</tbody>
</table>

Table 1 – Accuracy results of training set and validation set
Figure 8 – Autocorrelation check for forecast errors

Figure 9 – Residual checks for forecast errors
A good way to test whether the Holt-Winter method, which was used here, is an adequate predictive model for export volumes of the ports is to use the autocorrelation function and Ljung-Box test. The correlogram results in Figure 8 show that there is little evidence of non-zero autocorrelations in the in-sample forecast errors. i.e. the autocorrelations do not exceed the significance bounds for lags 1-20. The p-values for Ljung-Box tests of (Port AU = 0.4675, Port CN = 0.04924) in Figure 10 also indicates that there is little evidence of non-zero autocorrelations at lags 1-20.

To visualize the distribution of forecast errors, a histogram is plotted in Figure 9 which shows that the distribution is more or less centered at zero, and is approximately normally distributed. Although there is a slight skew to the right for the Chinese port export data, the skew is relatively small and we can safely assume that forecast errors are normally distributed with mean zero. This implies that the prediction holds and the Holt-Winters exponential smoothing method we used provided an adequate model for export demand. The numerical results of the mean are given as below:

<table>
<thead>
<tr>
<th></th>
<th>Port AU</th>
<th></th>
<th>Port CN</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Forecast Error Mean</td>
<td>-0.024</td>
<td>Forecast Error Standard Deviation</td>
<td>0.230</td>
<td>Forecast Error Standard Deviation</td>
</tr>
<tr>
<td>Ljung-Box Test p-value</td>
<td>0.4675</td>
<td></td>
<td></td>
<td>Ljung-Box Test p-value</td>
</tr>
</tbody>
</table>

Figure 10 – Numerical results for testing Forecast Errors

The top-down analysis provides us with a good forecast which we can safely decompose into a per-voyage forecast. Voyages in our studies are periodic and are either weekly or fortnightly. There are special port calls which are tri-weekly but these form the minority as it is important for a carrier to establish a shipping schedule that is highly predictable. To enable predictable port calls, carriers sometimes deploy up to 4 vessels on the same route cycle.
3.6.2 Bottom-Up Time Series Analysis

Thus far, we have approached the forecast from Top-down, on an aggregated basis. From our tests, the forecast results are pretty reasonable and we will use these results to calculate our per-voyage volume forecasts. However, there are times where a more detailed forecast breakdown is required. For instance, we may need the breakdown by port pairs which will help us predict the utilization of the vessels and also for other purposes like calculating the cycle stock where we need port pair-wise export volumes and their corresponding transit time.

This can be performed by collating forecasted figures in demand plans, and firmed demands from early shipment booking, and other customers' supplied information like customers' export forecast and volume commitment. Marketing, Sales and Operations can also provide inputs or special events which will alter the demand level. All information can be obtained from a carrier's Liner/Enterprise Resource Planning System (ERP) and Customer Relationship Management (CRM) systems. Simple port-wise allocation for origin and destination can then be done using the forecasted figures. The allocation will be augmented by the firmed demands and information provided by Marketing, Sales and Operations.
Chapter 4

4.1 Customer Service Level and Safety Stock

Empty containers are provided by the carrier to customers for shipments. An inventory of containers is maintained to satisfy customers' bookings. Stockouts of container inventory can lead to negative repercussions. These include decreased long-term customer satisfaction level, lost sales and general decreased perception of quality of service provided by the carrier. Uncertainties in supply and demand are the main factors that contribute to a shortfall in the inventory. There is therefore a need to have buffer inventory or otherwise known as safety stock. The volume of safety stock a carrier keeps in its global inventory can dramatically affect the business. Unutilized empty containers incur storage costs while shortfalls incur leasing costs. A carrier should maintain an optimum level of safety stock in the assigned depots or container yards which is adequate to meet customers' demand. The next section explains how we come up with an inventory model to calculate the safety stock level that is required based on the accuracy of our forecast figures. The safety stock is then incorporated in the base stock level optimization that we calculated to achieve a balance between shortage (leasing) and excess (storage) costs.
Figure 11 is a recap of what was discussed in the previous chapter and how it leads us to the calculation of safety stock which is an important consideration in both laden and empty container logistics. The Trend cum Seasonality analysis and forecast were first obtained in the time series analysis. The percentage error in the forecast is then calculated based on validation with historical data. This gives us the confidence interval of our forecast figures. Finally, to account for the uncertainty and be able to meet the demand with confidence, the buffer stock level is calculated. It is worth to note that we want to optimize the amount of
safety stock which has detrimental effect on a carrier's cost management in terms of 
purchasing and leasing containers and operational cost incurred due to storage of containers.

A naïve model with a constant forecast mean and variance will not work well here as there is 
seasonal variability and trend observed in the laden shipment demand. There are also months 
where the demand volatility is higher. We therefore try a tighter fit to the demand data by 
breaking down the mean and variance into months, giving us a multi-period inventory model. 
We also decompose the forecast into its major components which are export and import 
demand per day and lead time (interval between vessel arrivals). Lead time plays an 
important role in determining the safety stock level as an increase of a day in lead time means 
another day of demand has to be accounted for. Lead time can be affected by the following 
uncertain factors:

1) Ocean transit uncertainty
2) Port handling and customs hold uncertainty
3) Empty containers origin uncertainty
4) Empty containers supply uncertainty

It is noted in our literature review that lead time usually has a heavier right skew (Das, 2013). 
However, we found in our model validation that assuming a normal distribution is sufficient 
to give us a good approximation to our safety stock and base stock calculation.
4.3 Inventory Model and Safety Stock Calculation

In this section, we create a model for our inventory where we calculate the base stock level and safety stock. The reason why a base-stock model is used is because we have fixed time between orders (ie. Intervals between vessel arrivals). The review period is thus dictated by this ordering process consideration. The order quantity is as such variable. The base stock level which indicates the inventory position which an order quantity should satisfy is denoted by $B$ here.

\[ B = \text{mean demand} + z \cdot (\text{std dev}) \]

where \( z \) is the service level

Using the mean and variance of demand over transit time derivation proposed by Eppen & Schrage (1981), the equation becomes:

\[ \Rightarrow B = \mu_D \cdot \mu_L + z \sqrt{\mu_L \sigma_D^2 + \mu_D^2 \sigma_L^2} \quad (1) \]

where
- \( B \) is base-stock level
- \( \mu_D \) is the mean of per-day demand for containers
- \( \sigma_D \) is the standard deviation of the per-day demand for containers
- \( \mu_L \) is the mean of lead time for container replenishment where lead time is the interval between import vessel calls, which has a lot of uncertainty in some points. If vessel schedule does not have any variance, Equation (1) will reduce to \( B = \mu_D \cdot \mu_L + z \sqrt{\mu_L \sigma_D^2} \)
- \( \sigma_L \) is the standard deviation of the lead time for container replenishment
- \( z \) is the service level

Demand for containers is the net of export demand with the supply of empty containers from empty returns. We assume independence of the demand and supply variables.
\[ \mu_D = \mu_E - \mu_I \]

where

- \( \mu_E \) is the mean per-day export demand
- \( \mu_I \) is the mean per-day import demand

\[ \sigma_D = \sqrt{\sigma_E^2 + \sigma_I^2} \]

where

- \( \sigma_E \) is the export per-day standard deviation
- \( \sigma_I \) is the import per-day standard deviation

\[ B = (\mu_E - \mu_I) \cdot \mu_L + z \sqrt{\mu_L \cdot (\sigma_E^2 + \sigma_I^2) + (\mu_E - \mu_I)^2 \sigma_L^2} \]

\[ \Rightarrow B = (\mu_E - \mu_I) \cdot \mu_L + z \sqrt{\mu_L \cdot (\sigma_E^2 + \sigma_I^2) + (\mu_E - \mu_I)^2 \sigma_L^2} \]

We then extend it to a multi-period Inventory reorder point, where \( t \) denotes the review time. The review period is according to vessel schedule, i.e., Time interval between consecutive vessel calls.

\[ \mu_{Dt} = \mu_{Et} - \mu_{It} \]

\[ \sigma_{Dt} = \sqrt{\sigma_{Et}^2 + \sigma_{It}^2} \]

\[ \Rightarrow B_t = \mu_{Dt} \cdot \mu_{Lt} + z \sqrt{\mu_{Lt} \cdot (\sigma_{Et}^2 + \sigma_{It}^2) + (\mu_{Et} - \mu_{It})^2 \sigma_{Lt}^2} \]

\[ \Rightarrow B_t = (\mu_{Et} - \mu_{It}) \cdot \mu_{Lt} + z \sqrt{\mu_{Lt} \cdot (\sigma_{Et}^2 + \sigma_{It}^2) + (\mu_{Et} - \mu_{It})^2 \sigma_{Lt}^2} \] (2)

Since we are forecasting the safety stock for future, we will simply use the forecast figures \( \hat{\mu}_{Et} \) and \( \hat{\mu}_{It} \) as the export and import means respectively. The standard deviation is then the RMSE of the export and import forecasts, \( \hat{\sigma}_{Et} \) and \( \hat{\sigma}_{It} \) respectively.

\[ \Rightarrow B_t = (\hat{\mu}_{Et} - \hat{\mu}_{It}) \cdot \mu_{Lt} + z \sqrt{\mu_{Lt} \cdot (\hat{\sigma}_{Et}^2 + \hat{\sigma}_{It}^2) + (\hat{\mu}_{Et} - \hat{\mu}_{It})^2 \sigma_{Lt}^2} \] (3)
4.4 Service Level Monte Carlo Simulation

Based on the model that we have created, we performed a Monte Carlo simulation on Port AU and CN's data to check the accuracy of our model. The base stock level is calculated as per Equation (3) in the previous section. The results are as shown below:

Mean Inventory and Safety Stock for a Short Transit Port (Schedule = 14 days) for Port AU. Schedule Coefficient Variance = 0.27

<table>
<thead>
<tr>
<th>Target Service Level</th>
<th>Actual Service Level</th>
<th>Mean Inventory Level</th>
<th>Safety Stock</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>0.74</td>
<td>345</td>
<td>140</td>
</tr>
<tr>
<td>0.85</td>
<td>0.85</td>
<td>378</td>
<td>173</td>
</tr>
<tr>
<td>0.9</td>
<td>0.89</td>
<td>419</td>
<td>214</td>
</tr>
<tr>
<td>0.95</td>
<td>0.98</td>
<td>479</td>
<td>274</td>
</tr>
<tr>
<td>0.98</td>
<td>0.99</td>
<td>547</td>
<td>342</td>
</tr>
<tr>
<td>0.99</td>
<td>1.00</td>
<td>593</td>
<td>388</td>
</tr>
</tbody>
</table>

Table 2 - Service level and Inventory simulation for Port AU

Mean Inventory and Safety Stock for a Short Transit Port (Schedule = 6-7 days) for Port CN. Schedule Coefficient Variance = 0.46

<table>
<thead>
<tr>
<th>Target Service Level</th>
<th>Actual Service Level</th>
<th>Mean Inventory Level</th>
<th>Safety Stock</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>0.90</td>
<td>134</td>
<td>19</td>
</tr>
<tr>
<td>0.85</td>
<td>0.91</td>
<td>139</td>
<td>24</td>
</tr>
<tr>
<td>0.9</td>
<td>0.93</td>
<td>144</td>
<td>29</td>
</tr>
<tr>
<td>0.95</td>
<td>0.95</td>
<td>153</td>
<td>38</td>
</tr>
<tr>
<td>0.98</td>
<td>0.95</td>
<td>162</td>
<td>47</td>
</tr>
<tr>
<td>0.99</td>
<td>0.97</td>
<td>168</td>
<td>54</td>
</tr>
</tbody>
</table>

Table 3 - Service level and Inventory simulation for Port CN

The Target Service Level is the model's input target service level, and the multi-period Reorder point and Safety Stock is calculated from our model. The Actual Service Level is calculated as (1 - actual stock out percent). The Mean Inventory Level is the average stock level in the inventory over the course of a year.
The Monte Carlo simulation results in Table 2 and Table 3 show us that the model that we have described is accurate. Except for the target service level of 0.8, our model's target service level is usually met or the actual service level from the simulation is only less than 3% to meeting the target service level. It means that we can rely on the model for our inventory policy of safety stock and reorder point.

Our model also tells us that when schedule coefficient variance is high, the safety stock required will be higher. The inaccuracy of the schedule will also give rise to a poorer prediction of safety stock to meet target service level.

Compiling what we have discussed in this chapter, we define the process of Supply Network Planning as below:

<table>
<thead>
<tr>
<th>Supply Network Planning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Define the Inventory Model</td>
</tr>
<tr>
<td>2) Determine Customer Service Level</td>
</tr>
<tr>
<td>3) Calculate Safety Stock level</td>
</tr>
<tr>
<td>4) Determine Ordering Strategy</td>
</tr>
<tr>
<td>a. Base stock level ordering strategy</td>
</tr>
<tr>
<td>5) Determine possible sources of replenishment</td>
</tr>
<tr>
<td>a. Empty returns for import shipments</td>
</tr>
<tr>
<td>b. Empty Repositioning</td>
</tr>
<tr>
<td>c. Leasing</td>
</tr>
</tbody>
</table>
Chapter 5

5.1 Empty Container Repositioning

In the previous section, we have created empty container demand forecasts for ports in our network. To satisfy the demand for containers which we have forecasted, the options are 1) Empty repositioning (moving empty containers from surplus ports to deficit ports) and 2) Short term lease from container leasing companies. In addition, the replenishment and distribution plan must consider the costs that are incurred in both options. Empty container repositioning costs comprise of stevedore, pickup handling, trucking and feeder costs. Leasing of containers incurs lift-on life-off handling costs and higher leasing per diem costs for short term leases, and the associated costs are typically higher than empty repositioning and require more handling effort (surveying of containers etc. before use). As such, carriers would minimize leasing containers to optimize costs, and leasing is limited to prevent backorder which will cause delay in fulfilling customers' shipments. If a carrier is consistently leasing containers on short-term, it is a signal that the container fleet is running too lean and that decisions to purchase new containers to increase the fleet size should be considered.

There are a few complications in making decisions to perform empty repositioning. First, the cost of empty repositioning and leasing for each port is different. The demand for empty containers for each port is different across months i.e. A port, which has surplus containers in the current month, may need containers for shipments a few months later. The decision lies whether to empty reposition the empty containers to other ports or hold them for potential future use. There is therefore an opportunity to create an optimization model which can be used to assist the decision support process. The challenge is to create a distribution plan that
takes into account satisfying the demand requirements and lowering the cost at the same time.

Empty container logistics is a well-researched area and there are a few models which have been implemented in carriers (Epstein, Neely, & Weintraub, 2012). Mathematical programming is commonly used to model global empty container repositioning. Feng & Chang, (2008) formulated the mathematical programming problem into two stages where one is to estimate the empty container stock at each port and the other is to model the empty container reposition planning as the Transportation Problem which is then solved by Linear Programming. The authors also proposed the partition of shipping network into smaller geographical regions and reposition empty containers within the regions to avoid capacity lost on long-distance high revenue links. Wang & Jiang (2010) focused on optimizing the problem of empty container repositioning with multiple origins, intermediate and destination ports. They proposed a mixed integer linear programming model to determine the optimal location of different types of container among different shipping schedules. Hajeeh & Behbehani (2011) used a mathematical model to find an optimal sequence of ships' movement among ports in order to satisfy demands at the ports with minimum total costs. The original problem was transformed into Transportation Problem (TP) and Capacitated Plant Location Problem (CPL). The latter was used to find the optimal solution of the complete problem. Epstein et al. (2012) presented a multi-commodity network flow model with considering capacity and operational constraints.

To model uncertainty in the shipping industry, Lai (2013), Francesco et al. (2013) & Francesco et al. (2010) investigated repositioning policies and used multi-scenario model, where scenarios are linked by non-anticipativity conditions. The uncertainty comes in forms
of temporary events like port congestions, that prevent the empty containers from being timely repositioned for use. Results show that the multi-scenario solutions of demand fulfilment rate, for different evolutions caused by uncertainties in the ocean leg, can serve as a hedge against uncertainty and exhibit some forms of robustness by mitigating the risks of not meeting empty container demand. Another way to mitigate uncertainty is a robust optimization approach proposed by Frera et al (2009). In the solutions given, a feasible recovery plan exists for every joint realization of uncertain parameters. The premise was that only a predefined number of parameters take on their worst value.

Some literature extended the mathematical models and algorithms to beyond empty repositioning. Bandeira et al. (2009) proposed a Decision Support System for integrated distribution of empty and full containers and addressed the problem as a network of customers and suppliers like leasing companies. The model operates in stages by prioritizing laden containers before empty containers and then statistically optimizing costs. Lastly, Braekers et al. (2011) and Yur & Esmer (2011) conducted detailed literature reviews and compared different optimization methods and frameworks.

This thesis focuses on the study of the ocean freight optimization. There are also opportunities for cost cutting and improving efficiency on inland freight and trucking. This is a well-studied area and approaches, like street turns and other optimizations, have been studied and proposed in Jula et al (2003) and Le (2003). When using street turns, empty containers are trucked directly from consignees to shippers, without trucking back to an inland depot. This reduces the number of empty container movements and thus the cost involved. It also alleviates inland travelling and waiting time for congested ports by avoiding the terminals, reducing external costs of congestion even further (Dong & Song, 2009).
5.2 Leasing and Holding Empty Container Inventory

Containers are mostly purchased and owned in larger ocean carriers. A smaller portion of the containers may be rented or leased containers. Leasing containers from container leasing companies comes in several forms. They can be long term contracts, lease and purchase contracts or short term lease contracts. The container leasing adds flexibility to container fleet management as the empty containers can be picked up by carriers when there is an increased demand and returned to the leasing companies when the containers are not required anymore. Although container leases meet a carrier's need for an elastic fleet, they, however, are subjected to specific clauses which are indicated in rental contracts. There are usually limits to the rental period and also the return volume. In addition, there are extra operating costs generated by rented containers in the form of per-diem, container survey, trucking and other lift-on and lift-off charges. Per-diem costs are for example even higher for short-term leases. Although rented containers incur higher costs, it is worth investigating the need for leasing containers to prevent back-orders of containers which leads to lower customer satisfaction. A deterministic multi-commodity model was created by Moon et al. (2010) which took container purchasing and short-term leasing options into consideration.

Empty containers can be held in a location as long as there is sufficient storage capacity in depots. Depots can be provided by external companies or are self-owned. The former will incur costs which come in form of storage pool cost or storage days cost. The former is a tiered cost based on the number of containers stored in a depot facility per day and the latter is a flat per-day rate for stored containers which is billed fortnightly or monthly. It is important to optimize the utilization of the containers for laden shipments (which generates
revenue for a carrier) and reduce their depot stay due to incurrence of storage cost or space constraints. Containers which are left sitting in depots for undesirable length of time are called 'long stay' or 'overstayed' containers.

5.3 Problem Formulation

The empty container replenishment and export cycle is formulated as a network flow problem. The inflows of a node (ie. a port) in the network flow are empty containers returned by import consignees, empty repositioning into the port, and new short term leasing of containers. The outflows of a node consist of laden containers for export shipments, empty repositioning out of a node, offhire (returning of short-term lease containers) and sale of owned containers. See Figure 12 for an illustration of the flows of a port node \( P_1 \).

![Network Flow](image)

Figure 12 – Network Flow w.r.t. a Port \( P_1 \)
There are three mutually exclusive scenarios for each port: 1) Surplus Port (where there is a net outflow of containers due to import trade imbalance i.e. import volume is higher than export volume. Empty containers thus have to be moved out of the port), 2) Deficit Port (where there is a net inflow of containers due to higher export volume) and 3) Neutral Port (where there is trade balance). Figure 13 shows possible replenishment or distribution plans for Port P₁ depending on the type of port it is.

The model is formulated to reflect actual flow of containers as closely as possible. Below are the premises and safe assumptions we based our model on without loss of validity in the model.

Premises:

1) We consider only 20-footer general purpose containers here and the model can be extended to multi-commodity
2) We consider owned containers and long term lease-and-purchase containers as our pre-existing fleet.

3) Short term leased containers are used when there is a shortage of empty containers to meet demand requirements. There is usually a minimum leasing period imposed even on short-term lease and these containers are returned to the leasing companies whenever possible to reduce cost.

4) The off-hire cost (lift-on lift-off, trucking and washing etc) is associated with the on-hire process of leasing.

5) Although, we focus mainly on the ocean leg of the entire freight, we considered door to door services which include inland trucking and added the incurred transit delay in our model.

Assumptions:

1) There is no limit on the slot allocation on vessel for empty containers. This is a possible area for future work to extend the model and the assumption is valid for our case as we work with a carrier that has available slots on most empty repositioning routes and has flexibility in vessel deployment (most of the vessel fleet is leased). A future work is to extend the model to other carriers where a constraint has to be added for slot allocation for empty containers or alternative higher costs for using partner carriers have to be considered.

2) We do not set constraints on the holding capacity of the depots as the objective of the model is to understand the most optimal distribution plan with respect to cost. The space for storing containers can then be negotiated with depots and container yards.
3) To formulate an empty container repositioning problem, decision variables are typically number of containers and therefore are integer values. However this integer requirement increases modelling complexity and augments computation time. In our model, we relax the integrality constraints as often, results are found to be a good approximations (Feng & Chang, 2008)(Moon et al., 2010)

4) There are some minor flows like container maintenance and repair which are not depicted in the model for better clarity and generality and can be easily added as constants in objective function if required.
5.4 Just-in-Time Optimization Model

Definitions:

\( i, j \in (1, ..., P) \) where \( 1, ..., P \) are set of ports in the network

\( t \in (1, ..., 12) \) is the planning horizon in months

\( l \in (1, ..., 11) \) is the pre-emptive empty repositioning period in months i.e. \( l = 1 \)

means that empty repositioning is performed 1 month earlier than required month

\( T_{ij} \) is the transit time from depot at port \( i \) to depot at port \( j \)

\( d_{ij} \) is customer's detention time for shipment (i.e. Export detention time for shipper at port \( i \) and import detention time for consignee at port \( j \))

\( E_{ij}^t \) is export demand volume (in terms of boxes) from port \( i \) to port \( j \) at time \( t \) (in months) and is exogenously determined. Forecast figures are used to approximate the exogenous demand.

\( I_{ij}^t \) is import empty return volume at time \( t \) and is calculated using export demand volume \( E_{ij}^t \), transit time \( T_{ij} \) and detention time \( d_{ij} \)

\( C_{ri,j} \) is the cost of empty repositioning from port \( i \) to port \( j \), where \( i \) is the origin port, \( j \) is the destination port

\( C_l \) is the cost of short term leasing in port \( i \)

\( C_{hi} \) is the cost of Inventory holding in port \( i \)
Decision Variables:

\[ R_{ij}^t, L_i^t, O_i^t \]

\( R_{ij}^t \) is the decision variable to replenish port \( j \)'s inventory of empty containers by empty repositioning them from port \( i \) to port \( j \) where port \( i \) is likely to be a surplus area (with more units than safety stock required)

\( L_i^t \) is the alternative replenishment decision variable to supply empty containers by performing short-term lease (ie. number of units of empty containers leased at port \( i \) at time \( t \) in months)

\( O_i^t \) is the lease offhire count for surplus empty containers returned to leasing companies (ie. number of units of empty containers offhired at port \( i \) at time \( t \) in months). There is no associated cost for offhire as the cost (lift-on lift-off and trucking cost) is incorporated in the leasing cost.

Objective Function:

The objective function is to minimize the Total Relevant Cost (TRC):

\[
(1) \quad \min \left[ \sum_{t=1}^{12} \sum_{i=1}^{P} \sum_{j=1}^{P} R_{ij}^t \cdot C_{rij} + \sum_{t=1}^{12} \sum_{i=1}^{P} L_i^t \cdot C_{Li} \right]
\]

Constraints:

\[ \forall \ i \in (1, \ldots, P), \ t \in (1, \ldots, 12), \]

Assuming a consumption window of 1 month, we calculate the number of containers used across the different months that the consumption window may span across:

\[
R_{ij}^t = E_{ji}^{[t-T_{ji}-d_{ji}]} \cdot [1 - (\lfloor T_{ji} - d_{ji} \rfloor - (T_{ji} - d_{ji}))] + E_{ji}^{[t-T_{ji}-d_{ji}]} \cdot (\lfloor T_{ji} - d_{ji} \rfloor - (T_{ji} - d_{ji}))
\]

\[
R_{ji}^t = R_{ji}^{[t-T_{ji}]} \cdot [1 - (\lfloor T_{ji} \rfloor - T_{ji})] + R_{ji}^{[t-T_{ji}]} \cdot (\lfloor T_{ji} \rfloor - T_{ji})
\]
Conservation of flow gives us the following constraint:

\[
(2) \quad \sum_{j=1}^{P} -1(E_{ij}^t + R_{ij}^t) + I_{ij}^t + R'_{ij}^t = \sum_{j=1}^{P} L_i^t \geq 0
\]

Rearranging constraint (1) as below:

\[
(2a) \quad \sum_{j=1}^{P} R'^{rt}_{ij} - R_{ij}^t = L_i^t + O_i^t \geq \sum_{j=1}^{P} (E_{ij}^t - I_{ij}^t)
\]

Empty Repositioning Leasing RHS:
- Decision Variables
- Decision Demand

Other constraints:

\[
(2) \quad R_{ij}^t \geq 0, \quad R_{ij}^t \in \mathbb{Z}
\]

\[
(3) \quad P_{ij}^t \geq 0, \quad P_{ij}^t \in \mathbb{Z}
\]

\[
(4) \quad L_i^t \geq 0, \quad L_i^t \in \mathbb{Z}
\]

\[
(5) \quad i \neq j
\]

We can relax the above constraints to the below for a good approximation,

\[
(2a) \quad R_{ij}^t \geq 0, \quad R_{ij}^t \in \mathbb{R}
\]

\[
(3a) \quad P_{ij}^t \geq 0, \quad P_{ij}^t \in \mathbb{R}
\]

\[
(4a) \quad L_i^t \geq 0, \quad L_i^t \in \mathbb{R}
\]

This will give us a linear programming problem instead of an integer programming problem which is much faster to solve.

Another metric that we use to examine is Total Stock Level (Per month) $SL$ where
\[ SL_i^t = \sum_{j=1}^{P} \left[ R_{ij} \cdot T_{ij} + E_{ij}^t \cdot (T_{ij} + d_{ij}) \right] \]

\[ SL = \frac{1}{12} \sum_{t=1}^{12} \sum_{i=1}^{P} SL_i^t \]

5.5 Just-in-Time Optimization Results

To study the behavior of our model, we conducted a case study with data from the same company's network of 197 ports. The optimization process is executed at the start of year 2013 and forecast extrapolation uses training data from Year 2012 and before. We do not consider inputs from the Demand Plan Review stage in this optimization experimentation.

The optimized solution of Total Relevant Cost (TRC) that was produced was \textit{US$ 13,068,000} for the 20' GP container fleet. 23\% of the cost was leasing cost (\textit{US$ 3,019,000}). The optimized global container fleet is \textit{10,200} 20'GP boxes. The optimized distribution plans, for Port AU and Port CN, which were produced, are as below:

<table>
<thead>
<tr>
<th>Inbound Empty Repositioning Plan for Port AU</th>
<th>Inbound Empty Repositioning Plan for Port CN</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Repo Month</strong></td>
<td><strong>Planned Use Month</strong></td>
</tr>
<tr>
<td>Jan</td>
<td>Jan/Feb</td>
</tr>
<tr>
<td>Jan</td>
<td>Jan/Feb</td>
</tr>
<tr>
<td>Feb</td>
<td>Feb/Mar</td>
</tr>
<tr>
<td>Feb</td>
<td>Feb/Mar</td>
</tr>
<tr>
<td>Feb</td>
<td>Feb/Mar</td>
</tr>
<tr>
<td>Feb</td>
<td>Feb/Mar</td>
</tr>
<tr>
<td>Feb</td>
<td>Feb/Mar</td>
</tr>
<tr>
<td>Feb</td>
<td>Feb/Mar</td>
</tr>
<tr>
<td>Month</td>
<td>Month/Next</td>
</tr>
<tr>
<td>-------</td>
<td>-----------</td>
</tr>
<tr>
<td>Feb</td>
<td>Feb/Mar</td>
</tr>
<tr>
<td>Mar</td>
<td>Mar/Apr</td>
</tr>
<tr>
<td>Apr</td>
<td>Apr/May</td>
</tr>
<tr>
<td>Apr</td>
<td>Apr/May</td>
</tr>
<tr>
<td>Apr</td>
<td>Apr/May</td>
</tr>
<tr>
<td>Apr</td>
<td>Apr/May</td>
</tr>
<tr>
<td>Apr</td>
<td>Apr/May</td>
</tr>
<tr>
<td>Apr</td>
<td>Apr/May</td>
</tr>
<tr>
<td>Apr</td>
<td>Apr/May</td>
</tr>
<tr>
<td>Apr</td>
<td>Apr/May</td>
</tr>
<tr>
<td>May</td>
<td>May/Jun</td>
</tr>
<tr>
<td>May</td>
<td>May/Jun</td>
</tr>
<tr>
<td>May</td>
<td>May/Jun</td>
</tr>
<tr>
<td>May</td>
<td>May/Jun</td>
</tr>
<tr>
<td>May</td>
<td>May/Jun</td>
</tr>
<tr>
<td>Jun</td>
<td>Jun/Jul</td>
</tr>
<tr>
<td>Jun</td>
<td>Jun/Jul</td>
</tr>
<tr>
<td>Jun</td>
<td>Jun/Jul</td>
</tr>
<tr>
<td>Jun</td>
<td>Jun/Jul</td>
</tr>
<tr>
<td>Jun</td>
<td>Jun/Jul</td>
</tr>
<tr>
<td>Jun</td>
<td>Jun/Jul</td>
</tr>
<tr>
<td>Jul</td>
<td>Jul/Aug</td>
</tr>
<tr>
<td>Jul</td>
<td>Jul/Aug</td>
</tr>
<tr>
<td>Jul</td>
<td>Jul/Aug</td>
</tr>
<tr>
<td>Aug</td>
<td>Aug/Sep</td>
</tr>
<tr>
<td>Aug</td>
<td>Aug/Sep</td>
</tr>
<tr>
<td>Aug</td>
<td>Aug/Sep</td>
</tr>
<tr>
<td>Aug</td>
<td>Aug/Sep</td>
</tr>
<tr>
<td>Aug</td>
<td>Aug/Sep</td>
</tr>
<tr>
<td>Aug</td>
<td>Aug/Sep</td>
</tr>
<tr>
<td>Aug</td>
<td>Aug/Sep</td>
</tr>
<tr>
<td>Sep</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Sep</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Sep</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Sep</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Nov</td>
<td>Nov/Dec</td>
</tr>
<tr>
<td>Nov</td>
<td>Nov/Dec</td>
</tr>
<tr>
<td>Nov</td>
<td>Nov/Dec</td>
</tr>
<tr>
<td>Nov</td>
<td>Nov/Dec</td>
</tr>
<tr>
<td>Nov</td>
<td>Nov/Dec</td>
</tr>
<tr>
<td>Dec</td>
<td>Dec/Jan</td>
</tr>
<tr>
<td>Sep</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Sep</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Sep</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Oct</td>
<td>Oct/Nov</td>
</tr>
<tr>
<td>Oct</td>
<td>Oct/Nov</td>
</tr>
<tr>
<td>Nov</td>
<td>Nov/Dec</td>
</tr>
<tr>
<td>Nov</td>
<td>Nov/Dec</td>
</tr>
<tr>
<td>Dec</td>
<td>Dec/Jan</td>
</tr>
</tbody>
</table>
In Figure 14, *Repo Month* refers to the month where the reposition work order is issued. Due to transit time (ocean, inland transportation and port dwell), the empty containers may be used in the current month of the next month. We assume that the empty reposition volume per voyage is uniform and we set a consumption window of 1 month for the empty reposition. For example, if the *Repo Month* is January and Transit time is 0.4 month (ie. 12 days), the consumption window (highlighted in grey) is as below:

In the above example, *Planned Use Month* is therefore January and February (Jan/Feb as indicated in the distribution plan).
The optimization was performed with an Intel(R) Core(TM) i5 2.53GHz laptop with 8 GB of memory. The software optimizer that was used is Gurobi (academic version) (Gurobi Optimization, 2013).

5.6 Pre-emptive Repositioning and Inventory Holding

Optimization was performed for moving containers to satisfy customers' demand in a Just-in-time (JIT) manner. Just-in-Time inventory is good for supply chain where storage costs are higher. In our case, the storage cost for empty containers is relatively low, which means that we can relax the inventory hold a bit (ie. Holding inventory for a longer period of time) and explore the opportunities of lowering cost (by reducing empty repositioning etc.) To investigate that, we changed our model to include look ahead decision variables of sending empty containers pre-emptively.
Figure 16 shows an illustration of our model extension. In the extension, we not only add variables for pre-emptively sending empty containers, we considered allowing holding containers in the same location if there is surplus of containers which may be needed in the near future. As such at a particular point in time, decisions are made whether to send empty containers pre-emptively by \( \{1,\ldots,11\} \) months or hold inventory for \( \{1,\ldots,11\} \) months. The number of containers to send for each scenario is then modelled as individual decision variables.
Evidently, the solution as illustrated by one scenario shown in Figure 17 is a non-intuitive one as it involves a huge number of parameters such as availability, cost and demand requirements.

Figure 17 – Container Network Flow Space-Time Diagram

Figure 17 shows a space-time diagram of container flows. \( P_2^2 \) shows the Port \( P_2 \) at month 2. It is repositioning containers to Port \( P_3 \) which has empty container requirements for laden shipment denoted by variables \( E_{3,t} \). However, due to transit delay and delivery spanning across several voyages, the consumption window of the repositioned containers spans across month 2 and time 3. Variable \( R_{2,3}^2 \) denotes the number of containers that are repositioned.
Variable $S_{1,2}^{3,3}$ denotes pre-emptive sending (by 3 months earlier) of containers from Port $P_1$ to Port $P_2$ at month 1. The containers are therefore held at Port $P_2$ for 3 months before consumption. Variable $S_{1,1}^{2,2}$ denotes holding containers in the same location (i.e. Port $P_1$) for 2 months before the containers are used.
5.7 Pre-emptive Reposition Optimization Model

Definitions:
1 E (1, ..., M) is the pre-emptive empty repositioning period in months i.e. l = 1 means that empty repositioning is performed 1 month earlier than required month. M is an empty logistics management policy where M ∈ (1, ..., 11).

Decision Variables:
Rt., Sij, Lt, Oi

Sij is the alternative replenishment decision variable to satisfy empty containers demand by holding inventory (i = j) or preemptively performing empty containers from port i to j (where i ≠ j) l months earlier.

Objective Function:
The objective function then becomes the following with the additional pre-emptive empty repositioning term:

[1] \[ \min \left[ \left( \sum_{t=1}^{12} \sum_{i=1}^{P} \sum_{j=1}^{P} R_{ij}^t \cdot C_{rij} \right) + \left( \sum_{i=1}^{M} \sum_{t=1}^{12} \sum_{i=1}^{P} \sum_{j=1}^{P} S_{ij}^t \cdot Ch_i \right) + \left( \sum_{i=1}^{12} \sum_{i=1}^{P} L_i^t \cdot Cl_i \right) \right] \]

Constraints:
Conservation of flow gives us the following constraint:

[2] \[ \forall \ i \in (1, ..., P), \ t \in (1, ..., 12), \]

\[ \sum_{j=1}^{P} R_{ji}^t - R_{ij}^t + \sum_{i=1}^{M} \sum_{j=1}^{P} S_{ij}^t + L_i^t + O_i^t \leq \sum_{j=1}^{P} (E_{ij}^t - l_{ij}) \]

Empty Repositioning Decision Variables Pre-emptive Empty Repositioning Variable Leasing Decision Variable RHS: Demand constant
We need to add another constraint to ensure that there is no inventory holding in the same location unless there is surplus from import.

\[
\begin{cases}
0 \leq R_{it}^t \leq \sum_{j=1}^{P} (I_{ij}^t - E_{ij}^t) , & \text{if } \sum_{j=1}^{P} (I_{ij}^t - E_{ij}^t) < 0 \\
R_{it}^t = 0 , & \text{if } \sum_{j=1}^{P} (I_{ij}^t - E_{ij}^t) \geq 0
\end{cases}
\]

(3)

Total Stock Level SL is changed to include the holding inventory

\[
SL_i^t = \sum_{j=1}^{P} [R_{ij}^t \cdot T_{ij}^t + E_{ij}^t \cdot (T_{ij}^t + d_{ij})] + \sum_{l=1}^{M} \sum_{j=1}^{P} [S_{ji}^{(t-l)}, l] \cdot l
\]

(5)

\[
SL = \frac{1}{12} \sum_{t=1}^{12} \sum_{i=1}^{P} SL_i^t
\]

(6)

5.8 Pre-emptive Optimization Results

We ran the tests based on a range of pre-emptive months \(l = \{0, ..., 11\}\), where \(l=0\) is basically a JIT inventory. The results below in Figure 18 show that there are huge opportunities to reduce the Empty Repositioning cost and minimize leasing by allowing inventory holding. Containers in surplus areas can be sent to areas which need them pre-emptively. The optimization essentially makes sure that the areas with low repositioning costs (ie. nearer ports with lower stevedore costs) are used to replenish empty containers with global optimality. There is a potential cost savings of 23% comparing JIT to pre-emptive repositioning and inventory holding. It is seen from the results that the cost-saving benefit starts to decrease when the inventory holding policy is more than 2 months. The results thus suggest that a 2-month inventory holding policy is sufficient without the risk tradeoff of holding inventory for too long (where containers may not be required later).
We also looked at the total stock level that is required for each different inventory holding policy. The results show that with JIT, the size of container fleet required is much less and there is an increase of 90% of container fleet size requirements for longer inventory holding periods. The container fleet size calculations took into account the number of containers required in laden shipment and empty repositioning transit and inventory holding.

Figure 18 – Empty Logistics Cost against Inventory Holding Policy

Figure 19 – Stock Level against Inventory Holding Policy
We drill down to look at the cost savings that is brought about by the pre-emptive empty repositioning for Port AU and CN. We see from the results below that there are more opportunities for other supplier locations (with lower repositioning cost) and for supplier locations (with lower repositioning cost) to supply empty containers. Even though there is holding cost incurred (containers are sent early and are stowed at depots or container yards), the cost savings brought about by empty repositioning is still higher than that of holding cost which is relatively low (less than 4% of total relevant cost).

![Cost Breakdown by Empty Containers Supplier Country for Port AU](image)

Figure 20 – Cost Breakdown by Empty Containers Supplier Country for Port AU
Figure 21 – Cost Breakdown by Empty Containers Supplier Country for Port CN

Similarly, we produce the distribution plans for the two ports as below:

<table>
<thead>
<tr>
<th>Inbound Empty Repositioning Plan for Port AU</th>
<th>Inbound Empty Repositioning Plan for Port CN</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Repo Month</strong></td>
<td><strong>Repo Month</strong></td>
</tr>
<tr>
<td><strong>Planned Use Month</strong></td>
<td><strong>Planned Use Month</strong></td>
</tr>
<tr>
<td><strong>From Country</strong></td>
<td><strong>From Country</strong></td>
</tr>
<tr>
<td><strong>No. of Containers</strong></td>
<td><strong>No. of Containers</strong></td>
</tr>
<tr>
<td>Jan  Jan/Feb  AU *  166</td>
<td>Jan  Jan/Feb  TW  70</td>
</tr>
<tr>
<td>Jan  Feb/Mar  MY  134</td>
<td>Feb  Feb/Mar  TW  108</td>
</tr>
<tr>
<td>Feb  Feb/Mar  AU *  87</td>
<td>Mar  Mar/Apr  MY  176</td>
</tr>
<tr>
<td>Feb  Mar/Apr  MY  420</td>
<td>Mar  Mar/Apr  TW  26</td>
</tr>
<tr>
<td>Feb  Mar/Apr  BN  87</td>
<td>Apr  Apr/May  MY  252</td>
</tr>
<tr>
<td>Mar  Mar/Apr  AU *  104</td>
<td>Jun  Jun/Jul  MY  68</td>
</tr>
<tr>
<td>Mar  May/Jun  PG  33</td>
<td>Jul  Jul/Aug  MY  149</td>
</tr>
<tr>
<td>Apr  Apr/May  AU *  132</td>
<td>Aug  Aug/Sep  MY  220</td>
</tr>
<tr>
<td>Apr  May/Jun  MY  303</td>
<td>Aug  Aug/Sep  TW  281</td>
</tr>
<tr>
<td>Apr  May/Jun  BN  291</td>
<td>Sep  Sep/Oct  TW  29</td>
</tr>
<tr>
<td>Apr  May/Jun  PW  65</td>
<td>Oct  Oct/Nov  MY  69</td>
</tr>
<tr>
<td>Month</td>
<td>Month/Next Month</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------</td>
</tr>
<tr>
<td>May</td>
<td>May/Jun</td>
</tr>
<tr>
<td>May</td>
<td>Jun/Jul</td>
</tr>
<tr>
<td>May</td>
<td>Jun/Jul</td>
</tr>
<tr>
<td>Jun</td>
<td>Jun/Jul</td>
</tr>
<tr>
<td>Jun</td>
<td>Jul/Aug</td>
</tr>
<tr>
<td>Jun</td>
<td>Jul/Aug</td>
</tr>
<tr>
<td>Jul</td>
<td>Jul/Aug</td>
</tr>
<tr>
<td>Jul</td>
<td>Aug/Sep</td>
</tr>
<tr>
<td>Jul</td>
<td>Aug/Sep</td>
</tr>
<tr>
<td>Jul</td>
<td>Aug/Sep</td>
</tr>
<tr>
<td>Aug</td>
<td>Aug/Sep</td>
</tr>
<tr>
<td>Aug</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Aug</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Aug</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Sep</td>
<td>Oct/Nov</td>
</tr>
<tr>
<td>Sep</td>
<td>Oct/Nov</td>
</tr>
<tr>
<td>Sep</td>
<td>Oct/Nov</td>
</tr>
<tr>
<td>Sep</td>
<td>Sep/Oct</td>
</tr>
<tr>
<td>Nov</td>
<td>Dec/Jan</td>
</tr>
<tr>
<td>Nov</td>
<td>Nov/Dec</td>
</tr>
<tr>
<td>Dec</td>
<td>Jan/Feb</td>
</tr>
<tr>
<td>Dec</td>
<td>Feb/Mar</td>
</tr>
</tbody>
</table>

Ports are aggregated into countries here
* There are previous Australian port calls in the service route

Figure 22 – Inbound Distribution Plans with Pre-emptive Repositioning for Port AU and CN
5.9 Improving Optimization's Computation time

The preparation (creating objective and constraints) and computation time for the optimization model increase exponentially when the inventory holding policy period \( l \) increases. The computation at \( l = 9 \) takes more than 200 times longer to solve than \( l = 0 \). The table below shows the model suffers from the curse of dimensionality as the number of allowed holding periods increases.

<table>
<thead>
<tr>
<th>No. of Ports</th>
<th>Time Horizon</th>
<th>Inventory Holding Period</th>
<th>UB on No. of Decision Variables</th>
<th>No. of non-zero Decision Variables</th>
<th>No. of Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>197</td>
<td>12</td>
<td>1</td>
<td>931,416</td>
<td>64,932</td>
<td>4,728</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11</td>
<td>5,122,788</td>
<td>387,012</td>
<td>4,728</td>
</tr>
</tbody>
</table>

Table 4 – Computation complexity for different Inventory Holding Policy
We can relax the model’s constraints by removing constraint (3) from Section 0. This will reduce the model to only \( l = 1 \). The reduction is illustrated by the space-time diagram below.

![Space-Time Diagram](image-url)

**Figure 24 – Problem Reduction**

The decision variable \( S_{1,2}^{1,3} \) of performing a pre-emptive empty repositioning of 3 months earlier from port 1 to port 2 at time 1 is reduced to

\[
S_{1,2}^{1,3} = \{ S_{1,2}^{1,1}, S_{2,2}^{2,1}, S_{2,2}^{3,1} \}
\]

where \( S_{1,2}^{1,1} \) is the pre-emptive empty repositioning of 1 month earlier and \( S_{1,2}^{2,1}, S_{1,2}^{3,1} \) are decision variables of holding containers at port 2 for 1 month each at time 2 and 3. The set \( l \in (1, \ldots, 11) \) is thus reduced to \( l = 1 \). This will effectively give us a fast computation of the lower bound of cost (338 secs versus 2406 secs). The decision variable solution, however, does not indicate the holding period for empty repositioning.

The .NET source code for the Gurobi optimization discussed in Chapter 5 and Chapter 6 can be found in Appendix B.
5.10 Uncertainty Consideration and Robust Optimization

The model that was formulated in the previous sections considers only a single scenario of expected demand which is an output of the Demand Plan Review process. However, there is a confidence interval for the expected demand due to potential forecast errors caused by uncertainties in the ocean transportation. When we consider uncertainties, we need to take the range of possible outcomes (our objective cost function and total stock) into account and not assume the same expected outcomes. This is best explained by the “flaw of averages” which refers to a concept that it is not correct to calculate the average value of outcomes based on the average input conditions. It is associated with a simple mathematical proposition, as suggested by Neufville & Scholtes (2011), which states that:

The Average of all possible outcomes associated with uncertain parameters is generally not equal to the Value obtained from using the average value of the parameter. This is formally expressed as

\[ E[f(x)] \neq f[E(x)], \] except when \( f(x) \) is a linear function

The expression is sometimes called Jensen’s law and the mathematical proposition suggests that the outcome, from realistically formulated parameters with uncertainty considerations, generally differs from the outcome that one gets by considering a single scenario of average parameters.

The largest uncertainty in containers logistics, which contributes to the variability in demand for empty containers, can be attributed to shipment demands. Therefore, to understand the effects of demand uncertainty on our model, we did a sensitivity analysis by running 200
trials on normally distributed samples for individual port export demands. The uncertainty is
modelled as variance in normal distribution. By doing this sensitivity analysis, we are able to
obtain different distribution plans for effective repositioning of empty containers for the
different possible evolutions. We do not assign weightage to different scenarios and we use a
stochastic demand with uncertainties to reflect possible imbalance changes in the networks.
This is reflected by actual shipping data where imbalances change over the years.

The objective functions of Total Relevant Cost and total stock results are shown in Figure 25.
The results show that there is a potential increase in TRC of up to 17% and increase in fleet
size of up to 135%. The results show that the uncertainties must be accounted for especially
when accruing cost expenses and allocating vessel and depot space for empty containers.

![CDF of TRC with Export Demand CV](image1)

![CDF of Total Stock with Export CV](image2)

Figure 25 – Effect of Demand Variance on CDF of TRC and Total Stock
We hereby define the last process in our framework for the Supply Network Optimization process.

**Supply Network Optimization**

1) Define network flow model

2) Obtain input parameters for the model
   a. Indexed cost
   b. Export demands from Demand Plan review
   c. Transit time
   d. Customer detention time

3) Utilize mathematical programming to obtain optimal distribution with minimized Total Relevant Cost

4) Consider forecast error due to laden export demand uncertainties

5) Decision variables obtained from the optimum solutions (under different uncertainties scenarios) serve as a basis for Supply Distribution Plan
Chapter 6

6.1 Business Analytics and System Implementation

Business analytics is a term which refers to using data in context, processing it into information and deriving value from the information through techniques we have discussed so far. In our previous discussion, we have seen how predictive analytics (Forecasting) and prescriptive analytics (Optimization) can help minimize inventory cost in supply chain and logistics. The data from analytics model also provide us with insights into how different inventory policies can impact inventory cost. However, the work does not stop here. The model and optimization framework has to be deployed in practice and refined in several iterations to gather substantial value from it.

Figure 27 – Decision Support System Architecture for Liner Empty Logistics
To do that, the above software architecture for an empty container logistics decision support system is proposed. Similar decision support system and analytics engines have been proved to work in a commercial enterprise, which we have implemented for supply chain prescriptive analytics (Johnson, Lee, & Simchi-Levi, 2013). The proposed architecture, illustrated above, is appropriate to the generality of most carriers. In the proposed architecture, the module of Empty Logistics Optimizer (or MTLO in short) consists of the analytics’ forecasting and optimization engine. Data is first transferred from the main Enterprise Resource Planning system to MTLO’s database using Extract, Transform and Load process which can be performed with tools like IBM’s Informatica or Microsoft’s SQL Server Integration Services. The workflow engine within MTLO will then trigger the analytics process on a periodic basis, starting the forecasting process using a script (RScript in our case). The statistical tool used here is R or Revolution R Enterprise (with commercial license) that uses historical training data, from MTLO’s database, to obtain forecasted demands for the next planning period. The demand forecast is then reviewed in a collaborative process with information provided by Sales and Marketing (from contracts and volume commitment in the CRM system), Customer Service (early bookings and leads) and Empty Logistics Planners (forecast accuracy) etc. In addition, backward and forward consumption days have to be defined together with the forecast set. This creates a forecast consumption window that correspond to the sales order schedule, which can then be used to ensure that there is no duplicated demand count between forecasted / planned demands and firmed demands (early bookings).

The reviewed demand plan is then used as input to the final optimization engine which requires additional information such as transit times, customer detention times and lastly indexed costs of empty repositioning (Stevedore, handling, trucking, slot loss etc), leasing and
holding (storage cost) etc. The optimization engine’s objective is to minimize cost in our case. Based on our optimization results, the cost improvement can reach up to 31% compared to the actual costs. The optimization engine also produces an optimal distribution plan produced which can then serve as a reference or benchmark to the Empty Logistics Planners. The financial information can also be transferred to Finance for cost accrual estimation.

![Advantages of Business Analytics](image)

Adapted from: Davenport & Harris (2007)

Figure 28 – Competitive Advantage of Business Analytics

Evidently, there are many benefits that can be brought about by the use of business analytics especially in the area of predictive and prescriptive analytics. Figure 28 shows where the potential competitive advantage that a carrier stands to gain with a fully implemented system of our forecasting and optimization engine. We highlighted the areas that are achieved by our
6.2 Policy and Impact Discussion

With our models, we have shown that, there are a few tradeoffs that carriers have to balance, namely, empty repositioning versus leasing and just-in-time replenishment versus pre-emptive replenishment. In our models, we have used mathematical programming to achieve an optimal cost effective balance between empty repositioning and leasing. In our study, we have investigated the impact of the two different replenishment strategies and how they can affect total relevant cost and unconstrained container fleet size.

Figure 29 - Models against actual cost FY 2013

Figure 29 shows a comparison of the just-in-time, pre-emptive model (3 months) and actual costs. We see that because of relatively lower holding cost (storage for empty containers)
compared to the cost of active replenishment with empty repositioning and leasing, there are opportunities for carriers to save costs by holding containers for later use. The actual costs for the carrier that we studied show that such opportunities exist. Our understanding of the company is that there is currently no coordination between the empty repositioning team between different regions and each team has a myopic view of the demand and supply situation. They also do not coordinate with the forecasting team to obtain information which can help them in making replenishment decisions. As such, they are making knee-jerk decisions to move containers to whichever places that require replenishment. The inefficiency incurs unnecessary costs. The models in our proposed decision support system has incorporated optimization and forecasts to cut down on unnecessary moves which helps to reduce the costs. The distribution plans produced by the system also help to serve as a reference to the empty repositioning team for decision making.

![Models' Fleet Size against Actual FY 2013](image)

**Figure 30 - Models' fleet size against actual FY 2013**

Figure 30 shows the unconstrained fleet size as a result of the cost optimization. In our study, we did not impose constraints on the container fleet size. However, we may need to investigate further on how the constraints of the fleet size, limited by the capacity of our vessels and depot storage space, will affect the cost optimization objective. The actual fleet
size of the studied carrier suggests that although it is not holding inventory or pre-emptively sending empty containers for replenishment, her container fleet size is much larger than that of the JIT model. It suggests that there are idling (long stay) containers at the depot which is the situation that the carrier's management is trying to tackle.

6.3 Conclusion

Ocean carriers are traditionally concerned with their topline revenue. However, as the market is getting weaker in recent years, profit margins are getting smaller. Coupled with the sharp increase in cost like bunker costs, carriers now have to devote themselves to cost management in order to ensure profitability. Empty container logistics is a large component in the cost factor. Due to strong imbalances in trade economies, carriers tend to accumulate many containers in import oriented locations while suffering from shortage of containers to meet customers demand in export oriented locations. To correct the imbalance of containers' supply, carriers have to spend large amount of money to reposition their containers, lease short term containers or hold their containers for near future's use. Given the complexity of the network's imbalance situation and costs of distribution, logistics operations have to be assisted by business analytics and decision support systems which have predictive and prescriptive capability, which we have proposed in this thesis. The decision support and optimization systems can identify the most efficient use of limited resources and help carriers make important decisions faster and with more confidence. Operations can then be improved with increased efficiency and reduced risk.
6.4 Future Work

We recognize that there are additional optimization opportunities in the logistics processes. The thesis presented models based on single commodity (container size type). Although the models can be extended to multi-commodity in a trivial manner, there are opportunities to improve optimization by using equipment substitution. Since year 2011, Moshe Loberant of Maersk Line had been heading the operation efforts in utilizing non-operating reefers (NORs) where reefers are used as dry cargo containers to correct imbalances of reefer and dry cargo trade between Asia and South America (GCaptain, 2013). The associated cost savings were approximately USD $50 million in 2011 and the figures were expected to increase.

Another area of study would be the dynamics of empty container logistics. Thus far, this thesis focuses on static optimization; there are opportunities in study on the dynamics of how other policies will affect the objective of cost management separately and concurrently. Figure 3 shows a system dynamics model which can be used for future study on how multiple factors can affect the objective of minimizing cost in empty logistics.
Bibliography


Appendix A  - Forecasting Source Code in R

# General Settings
path = "C:\home\Dropbox\MIT\Research\Thesis\Data\"

# Export

# Read in vector
x <- scan(paste(path,"Forecast\Month_Actual_AU.txt",sep=""))
d <- density(x)
plot(d)
d

# Import

# Read in vector
x <- scan(paste(path, "Forecast\Monthly_IMPORT_AU.txt",sep=""))
d <- density(x)
plot(d)
d

# Schedule

# Read in vector
x <- scan(paste(path, "Forecast\Schedule.txt",sep=""))
d <- density(x)
plot(d)
d
mean(x)
sd(x)

# Train and Test sets

# Read data into time series (train and test set)
xts_all <- ts(x, frequency=12, start=c(2009,1))
xts_train <- ts(x[1:36], frequency=12, start=c(2009,1))
xts_test <- ts(x[37:56], frequency=12, start=c(2012,1))

# Forecast and test fit

# Apply multiplicative Seasonal
# Split data into training and test sets
xf_train <- ets(xts_train, model="MAM", damped=F)
xf_test <- ets(xts_test, model=xf_train)

# Graph
plot(xf_train)
plot(fitted(xf_train))

# Check the accuracy of the forecast
accuracy(xf_train)
accuracy(xf_test)

# Test fit of forecast
acf(xf_train$residuals, lag.max=20)
Box.test(xf_train$residuals, lag=20, type="Ljung-Box")

# Check if errors is normally distributed
plotForecastErrors(xf_train$residuals)
mean(xf_train$residuals)
sd(xf_train$residuals)

# Convert time series of training and testing data back to vector
x_array <- unclass(fitted(xf_train))
x_array <- c(x_array, unclass(fitted(xf_test)))

# Write to csv file
write.csv(x_array, file = paste(path, 'output.csv', sep=''))

# Function to plot histogram
# Library
# source: Little red book of r

plotForecastErrors <- function(forecasterrors)
{
    # make a histogram of the forecast errors:
    mybinsize <- IQR(forecasterrors)/4
    mysd <- sd(forecasterrors)
    mymin <- min(forecasterrors) - mysd*5
    mymax <- max(forecasterrors) + mysd*3

    # generate normally distributed data with mean 0 and standard deviation
    mynorm <- rnorm(10000, mean=0, sd=mysd)
    mymin2 <- min(mynorm)
    mymax2 <- max(mynorm)
    if (mymin2 < mymin) { mymin <- mymin2 }
    if (mymax2 > mymax) { mymax <- mymax2 }

    # make a red histogram of the forecast errors, with the normally
distributed data overlaid:
    mybins <- seq(mymin, mymax, mybinsize)
    hist(forecasterrors, col="red", freq=FALSE, breaks=mybins)
    # freq=FALSE ensures the area under the histogram = 1
    # generate normally distributed data with mean 0 and standard deviation
    mysd
    myhist <- hist(mynorm, plot=FALSE, breaks=mybins)
    # plot the normal curve as a blue line on top of the histogram of forecast
    errors:

points(myhist$mids, myhist$density, type="l", col="blue", lwd=2)
Appendix B  - Empty Logistics Optimizer Gurobi Source Code

using System;
using System.Linq;
using System;
using System.Linq;
using System.Data;
using System.IO;
using System.Collections.Generic;
using System.Collections;
using System.Diagnostics;
using System.Configuration;
using EnSupport.Base;
using ExcellLibrary;
using Gurobi;
using MathNet.Numerics.Distributions;
using EnSupport.DB;

namespace LogOptimizer
{
    public class LogModelMthExGurobi
    {
        public class Parameters
        {
            public int numDecisionVars;
            public int numPortPairsMonth; // i * j * t
            public int numPorts; // i
            public int numTrials;
            public string distribution;
            public double coeffVariance;
            public double[] demand_E;
            public double[] transitDays;
            public double[] detentionDays;
            public double[] cost_repo;
            public double[] cost_lease;
            public double[] cost_offhire;
            public double[] cost_hold;
            public String SolveType;
            public bool bVerbose;
            public int earlyRepoMonths;

            public Parameters()
            {
                numPortPairsMonth = -1; // i * j * t
                numPorts = -1; // i
                demand_E = null;
                transitDays = null;
            }
        }
    }
}
detentionDays = null;
cost_repo = null;
cost_lease = null;
cost_offhire = null;
cost_hold = null;
numTrials = -1;

earlyRepoMonths =
Convert.ToInt32(ConfigurationManager.AppSettings["EarlyRepoMonths"])+1;

public class Results {
    public int[] idList;
    public ArrayList elapsedTimes;
    public ArrayList objectives;
    public ArrayList stockPerMonthMean;
    public ArrayList stockPerMonthStdDev;
    public double expectedObjective;
    public String solveResults;
    public int numPortPairsMonth;
    public int numPorts;

    //public double[] repoin;
    public double[] repoout;
    public double[] cost_coeff;
    public double[] lease;
    public double[] offhire;

    public Results() {
        idList = null;
        //repoin = null;
        repoout = null;
        lease = null;
        offhire = null;

        objectives = new ArrayList();
        stockPerMonthMean = new ArrayList();
        stockPerMonthStdDev = new ArrayList();
        elapsedTimes = new ArrayList();

        expectedObjective = double.MinValue;
        solveResults = "";
    }
}

Parameters m_params = new Parameters();
bool m_bLogToFile = false;
bool m_bSaveDetails = false;
StreamWriter m_fs;
Results m_results;
String m_filename;
MersenneTwister m_rnd;

int errCount = 0;
public LogModelMthExGurobi()
{
    String t = ConfigurationManager.AppSettings["LP_IP_Type"];
    m_params.SolveType = t;

    int numTrials = Convert.ToInt32(ConfigurationManager.AppSettings["NumTrials"]);
    if (m_params.numTrials == -1) m_params.numTrials = numTrials;
    m_params.distribution = ConfigurationManager.AppSettings["Distribution"].ToUpper(); // CONSTANT | NORMAL

    m_bLogToFile = false;
    m_fs = null;
    m_params.bVerbose = false;

    m_results = new Results();

    if (m_params.coeffVariance <= 0.0)
        m_params.coeffVariance = Convert.ToDouble(ConfigurationManager.AppSettings["CoefficientVariance"]);

    m_filename = "";
}

public LogModelMthExGurobi(String logFile,
    String solveType, /* "INTEGER" | "LINEAR" */
    double cv,
    int emptyRepoMonths,
    bool bSaveDetails,
    int seed
    ) : this()
{
    m_bLogToFile = true;
    m_bSaveDetails = bSaveDetails;
    m_fs = new StreamWriter(logFile);
    m_params.coeffVariance = cv;

    if (solveType == "INTEGER" ||
        solveType == "LINEAR")
    {
        m_params.SolveType = solveType;
    }

    m_params.earlyRepoMonths = emptyRepoMonths + 1;

    if (seed > 0)
    {
        m_rnd = new MersenneTwister(seed);
    }
    else
    {
        m_rnd = new MersenneTwister();
    }
}

~LogModelMthExGurobi()
{
    if (m_fs != null)
    {
// m_fs.Close();
  m_fs = null;
}  
m_bLogToFile = false;
}

public void Close()
{
  if (m_fs != null)
  {  
    m_fs.Close();
    m_fs = null;
  }
  m_bLogToFile = false;
}

private void LogWriteLine(String s, params Object[] obs)
{
  if (m_bLogToFile)
  {
    m_fs.WriteLine(String.Format(s, obs));
  }
  else
  {
    Console.WriteLine(String.Format(s, obs));
  }
}

private void LogWrite(String s, params Object[] obs)
{
  if (m_bLogToFile)
  {
    m_fs.Write(String.Format(s, obs));
  }
  else
  {
    Console.Write(String.Format(s, obs));
  }
}

public void SolveModel(String filename, bool bWaitForInput)
{
  DataSet ds = null;

  m_filename = filename;
  if (filename.EndsWith(".csv"))
  {
    int idx = filename.LastIndexOf(".");
    String filenameWoExt = filename.Substring(0, idx);
    ds = DataSetHelper.LoadFlatfile(new String[] { filenameWoExt + ".csv", filenameWoExt + ".LeaseAndHold.csv" }, "CSV");
  }
  else
  {
    ds = DataSetHelper.LoadExcel(filename);
  }
}
if (ds != null) SolveModel(ds, bWaitForInput);
}

private Hashtable GetColMap(DataSet ds)
{
    Hashtable ht = new Hashtable();
    if (ds.Tables.Count >= 1)
    {
        for (int i = 0; i < ds.Tables[0].Columns.Count; i++)
        {
            String colName =
            DataSetHelper.GetString(ds.Tables[0].Columns[i].ColumnName);
            ht[colName] = i;
        }
    }
    return ht;
}

public void SolveModel(DataSet ds, bool bWaitForInput)
{
    bool bVerbose = mparams.bVerbose;
    int batchId = Convert.ToInt32(String.Format("{0:MMddHHmmss}", DateTime.Now));

    // Let's calculate the time required
    Stopwatch timer = new Stopwatch();
    timer.Start();
    if (ds.Tables.Count > 0 &&
        ds.Tables[0].Rows.Count > 0)
    {
        DebugHelper.Print("[LogModelMthExGurobi::SolveModel] Data loaded
        successfully");

        // Read data, Table 0 contains the port pairs, demand volume and cost of
        repositioning
        // Table 1 contains the ports and the associated leasing cost
        int numPortPairsMonth = ds.Tables[0].Rows.Count;
        int numPorts = ds.Tables[1].Rows.Count;

        // Reset results
        m_results = new Results();

        // The following are the constants / coefficients from data
        m_params.cost_repo = new double[numPortPairsMonth *
        m_params.earlyRepoMonths];
        m_params.transitDays = new double[numPortPairsMonth];
        m_params.detentionDays = new double[numPortPairsMonth];
        m_params.demand = new double[numPortPairsMonth];

        // Set up the constants / coefficients in an array for the setup of
        objective function and
        // constraints later
        int monthIdx = DataSetHelper.GetColIdx(ds, "Month");
        int costRepoIdx = DataSetHelper.GetColIdx(ds, "Cost_Repo");
        int TransitDaysIdx = DataSetHelper.GetColIdx(ds, "TransitDays");
        int ExportDemandIdx = DataSetHelper.GetColIdx(ds, "ExportDemand");
int DetentionDaysIdx = DataSetHelper.GetColIdx(ds, "DetentionDays");

for (int k = 0; k < ds.Tables[0].Rows.Count; k++)
{
  int month = DataSetHelper.GetValue<int>(ds, monthIdx, k);
  double costRepo = DataSetHelper.GetValue<double>(ds, costRepoIdx, k);
  double transit = DataSetHelper.GetValue<double>(ds, TransitDaysIdx, k);
  double demand = DataSetHelper.GetValue<double>(ds, ExportDemandIdx, k);
  double detention = DataSetHelper.GetValue<double>(ds, DetentionDaysIdx, k) % (numPorts * numPorts);

  for (int l = 0; l < m_params.earlyRepoMonths; l++)
  {
    m_params.cost_repo[k + l*numPortPairsMonth] = costRepo;
  }
  m_params.transitDays[k] = transit;
  m_params.detentionDays[k] = detention;
  m_params.demandE[k] = demand;
}

m_params.cost_lease = new double[numPorts];
// Offire cost is incorporated in Onhire
// m_params.cost_offhire = new double[numPorts];
m_params.cost_hold = new double[numPorts];
for (int k = 0; k < ds.Tables[1].Rows.Count; k++)
{
  double costLease =
  DataSetHelper.GetValue<double>(ds.Tables[1].Rows[k][1]);
  double costHold =
  DataSetHelper.GetValue<double>(ds.Tables[1].Rows[k][2]);
  // double costOffHire =
  DataSetHelper.GetValue<double>(ds.Tables[1].Rows[k][3]);
  m_params.cost_lease[k] = costLease;
  m_params.cost_hold[k] = costHold;
  // m_params.cost_offhire[k] = costOffHire;
}

// Total number of decision variables is num of port pairs (276 ports ^ 2) + num Ports (276 ports ^ 2)
// Decision variables = Reposition volume, Ri,j (where i,j is the set of ports) and Leasing volume, Li (where i is set of ports) and -Li (Offhire volume)
// m_params.numDecisionVars = numPortPairsMonth * m_params.earlyRepoMonths + numPorts * 12 * 2;
// m_params.numPorts = numPorts;
// m_params.numPortPairsMonth = numPortPairsMonth;

if (bVerbose) LogWriteLine("\nLoad data success for i\*j\*t=(0), i={1}",
  m_params.numPortPairsMonth, m_params.numPorts);

// Solve the model
if (bVerbose) LogWriteLine("Solving...");
Solve();

DebugHelper.Print("[LogModelMthExGurobi::SolveModel] Solved successfully");
timer.Stop();
LogWriteLine("Elapsed time: \{0\} min \{1\} s",
Convert.ToInt32(timer.ElapsedMilliseconds / 60 / 1000),
Convert.ToInt32(timer.ElapsedMilliseconds / 1000 % 60));

//
LogWriteLine("Repo_out(m=0), Repo_out(m=1), Repo_out(m=2), Repo_out(m=3), Lease(Year), Offhire (Year), Solution(s), Avg Stock");

DebugHelper.Print("[LogModelMthExGurobi::SolveModel] Saving results to
database");

// Write headers and trial results
for (int x = 0; x < m_params.numTrials; x++)
{
    AddResultsHeader( batchId,
        m_filename,
        m_params.coeffVariance,
        m_params.earlyRepoMonths - 1,
        (Int)m_results.elapsedTimes[x],
        x,
        (Double)m_results.objectives[x],
        (Double)m_results.stockPerMonthMean[x],
        (Double)m_results.stockPerMonthStdDev[x],
        DateTime.Now);
}

if (m_bSaveDetails)
{
    // Output the line results to file, only one instance as there are
too much data!
    for (int x = 0; x < m_results.repo_out.Length; x++)
    {
        int i = 0, j = 0, t = 0, l = 0;
        GetIdx(x, ref i, ref j, ref t, ref l);

        if (m_results.repo_out[x] > 0.0)
        {
            AddResultsLine(batchId,
                0,
                i, j, t, l,
                "REP",
                m_results.repo_out[x],
                0,
                m_results.cost_coeff[x]);
        }
    }

    for (int x = 0; x < m_params.demand_E.Length; x++)
    {
        int i = 0, j = 0, t = 0, l = 0;
        GetIdx(x, ref i, ref j, ref t, ref l);

        if (m_params.demand_E[x] > 0.0)
        {
            AddResultsLine(batchId,
                0,
                i, j, t, l,
                "EXP",
                m_results.demand_E[x],
                0,
                m_results.cost_coeff_E[x]);
        }
    }
}
m_params.demand_E[x], 0, -1;
AddResultsLine(batchId, 0, i, j, t, 0,
"TRD",
m_params.transitDays[x], 0, -1);
}
}

for (int x = 0; x < m_results.lease.Length; x++)
{
    if (m_results.lease[x] > 0.0)
    {
        AddResultsLine(batchId, 0, x % m_params.numPorts, -1, x / m_params.numPorts, -1,
"ONH",
m_results.lease[x], 0, -1);
    }
    if (m_results.offhire[x] > 0.0)
    {
        AddResultsLine(batchId, 0, x % m_params.numPorts, -1, x / m_params.numPorts, -1,
"OFH",
m_results.offhire[x], 0, -1);
    }
}
}

private void AddResultsHeader(int batchId, String Type, double CV, int EarlyRepoMonth,
int elapsedTime, int trialIdx, double objective,
double stockPerMonthMean, double stockPerMonthStdDev,
DateTime dt)
```csharp
private void AddResultsLine(int batchId, int trialIdx, int i, int j, int t, int l, String type, double valueFloat, int valueInt, double costCoeff)
{
    try
    {
        new DBHelper(DBHelper.DBType.LOG).ExecUpdStoredProc("AddResultsLine", new String[]
        {
            "@BatchId",
            "@TrialIdx",
            "@PortI",
            "@PortJ",
            "@Month",
            "@PreemptMth",
        },
        new object[]
        {
            batchId,
            trialIdx,
            i,
            j,
            t,
            l,
            type,
            valueFloat,
            valueInt,
            costCoeff,
        }
    }
    catch (Exception e)
    {
        errCount++;
        if (errCount < 5)
        {
            DebugHelper.Print("[LogModelMthExGurobi::SolveModel] Error: {0}",
             e.Message);
        }
    }
}
```
"@Type",
"@ValueFloat",
"@ValueInt",
"@CostCoeff"
},
    new object[] {
        batchId, 
        trialIdx, 
        i, 
        j, 
        t, 
        l, 
        type, 
        valueFloat, 
        valueInt, 
        costCoeff 
    }
); 
} 
catch (Exception e) 
{
    errCount++; 
    if (errCount < 5) 
    
    DebugHelper.Print("[LogModelMthExGurobi::SolveModel] Error: {0}", 
        e.Message);
    
} 
} 
private void CopyArray(ref double[] src, ref double[] dest, int offset) 
{
    if (src == null || src.Length <= 0) return; 
    if (dest == null) 
    
    dest = new double[src.Length + offset];
    } else if (dest.Length != src.Length + offset) 
    
    dest = null; 
    dest = new double[src.Length + offset];
    
    src.CopyTo(dest, offset); 
} 
private void PrintArray(ref double[] src, String prefix, String suffix) 
{
    LogWriteLine(ArrayToStr(ref src, prefix, suffix));
} 
private String ArrayToStr(ref double[] src, String prefix, String suffix) 
{
    String retStr = "";
    if (src == null) return "";
    retStr += (prefix + "[");

for (int i = 0; i < src.Length; i++)
{
    retStr += String.Format("{0}{1}", i != 0 ? ", " : ",
src[i]);
}
retStr += ("]" + suffix);

return retStr;
}
public void GetIdx(int idx, ref int i, ref int j, ref int t, ref int l)
{
    l = (int)Math.Floor((double)(idx / m_params.numPortPairsMonth));
    t = (int)Math.Floor((double)(idx %
m_params.numPortPairsMonth) / (m_params.numPorts * m_params.numPorts));
    j = (int)Math.Floor((double)(idx % m_params.numPortPairsMonth) %
(m_params.numPorts * m_params.numPorts) % m_params.numPorts);    
    i = (int)Math.Floor((double)(idx % m_params.numPortPairsMonth) %
(m_params.numPorts * m_params.numPorts) / m_params.numPorts);
}
public int GetIdx(int i, int j, int t, int l)
{
    if (t < 0) t += 12;
    if (t >= 12) t -= 12;
    int idx = t * m_params.numPorts + m_params.numPorts;
    idx += i * m_params.numPorts + j;
    idx += l * m_params.numPortPairsMonth;

    return idx;
}
private void GetAverageStockPerMonth(ref GRBVar[] vars, ref double average, ref double stdev)
{
    // Get the average containers over months
    double[] mthAvgFlow = new double[12];
    double avgOutflow = 0.0;
    double maxMthOutflow = 0.0;

    for (int t = 0; t < 12; t++)
    {
        int leasePortsPeriod = t * m_params.numPorts * 2;
        for (int l = 0; l < m_params.earlyRepoMonths; l++)
        {
            for (int i = 0; i < m_params.numPorts; i++)
            {
                // Total containers in transit / holding
                if (l == 0)
                {
                    mthAvgFlow[t] += vars[GetIdx(i, j, t, l)].Get(GRB.DoubleAttr.X) * 1.0 *
(m_params.transitDays[GetIdx(i, j, t, 0)]) / 30.0;
                    mthAvgFlow[t] += m_params.demandE[GetIdx(i, j, t, 0)] *
(m_params.transitDays[GetIdx(i, j, t, 0)]) +
m_params.detentionDays[GetIdx(i, j, t, 0)]) / 30.0;
```
} else {
    double o = vars[GetIdx(i, j, t, l)].Get(GRB.DoubleAttr.X);
    for (int li = 1; li <= 1; li++)
    {
        if (li >= 1)
        {
            mthAvgFlow[(t + li) % 12] += o;
        }
        mthAvgFlow[t] += o;
    }
}
// Deduct Offhire
mthAvgFlow[t] -= vars[m_params.numPortPairsMonth * m_params.numPorts + i +
leasePortsPeriod].Get(GRB.DoubleAttr.X);
}
avgOutflow += mthAvgFlow[t];
if (mthAvgFlow[t] > maxMthOutflow) maxMthOutflow = mthAvgFlow[t];
}
avgOutflow /= 12.0;
average = Statistics.Mean(mthAvgFlow);
stdev = Statistics.StandardDeviation(mthAvgFlow);
}

public void Solve()
{
    Stopwatch timer = new Stopwatch();
timer.Start();
double[] decisionVar = new double[m_params.numDecisionVars];
for (int i = 0; i < m_params.numDecisionVars; i++)
    { decisionVar[i] = 99999999.0;
    }

    GRBEnv env = new GRBEnv(); // new GRBEnv("LogModel.log");
    env.Set(GRB.IntParam.OutputFlag, 0);
    GRBModel model = new GRBModel(env);
    int numPorts = m_params.numPorts;
    int numPortPairsMonth = m_params.numPortPairsMonth;
    Console.WriteLine("Number of port pairs month={0}, number of ports={1}...",
numPortPairsMonth, numPorts);
    // Set Objective function for repo variable
    for (int l = 0; l < m_params.earlyRepoMonths; l++)
    {
        for (int t = 0; t < 12; t++)
            {
```csharp
int period = t * m_params.numPorts * m_params.numPorts;
for (int i = 0; i < m_params.numPorts; i++)
{
    for (int j = 0; j < m_params.numPorts; j++)
    {
        if (i != j)
        {
            int idx = GetIdx(i, j, t, 1);
            decisionVar[idx] = m_params.cost_repo[GetIdx(i, j, 0, 0)] + (m_params.cost_hold[j] * 1);
        }
        else
        {
            if (l == 0)
            {
                // Holding Cost
                decisionVar[GetIdx(i, i, t, 1)] = 99999999.0;
            }
            else
            {
                // Holding Cost
                decisionVar[GetIdx(i, i, t, 1)] = m_params.cost_hold[i] * 1;
            }
        }
    }
}

// Set Objective function for leasing variable
for (int t = 0; t < 12; t++)
{
    int leasePortsPeriod = t * m_params.numPorts * 2;
    for (int i = 0; i < m_params.numPorts; i++)
    {
        decisionVar[numPortPairsMonth * m_params.earlyRepoMonths + i + leasePortsPeriod] = m_params.cost_lease[i];
        decisionVar[numPortPairsMonth * m_params.earlyRepoMonths + numPorts + i + leasePortsPeriod] = 0; // All costs are considered in m_params.cost_offhire[i];
    }
}

Console.Write("Set objective function..." would=

double[] lb = new double[m_params.numDecisionVars];
double[] ub = new double[m_params.numDecisionVars];
char[] types = new char[m_params.numDecisionVars];
string[] names = new string[m_params.numDecisionVars];
// Set Bounds
for (int i = 0; i < m_params.numDecisionVars; i++)
{
    lb[i] = 0.0;
    ub[i] = 5000.0;

    // Integer
    if (m_params.SolveType == "INTEGER")
    {
    
```
types[i] = GRB.INTEGER;
} 
else 
{
    types[i] = GRB.CONTINUOUS;
}
}

GRBVar[] vars = model.AddVars(lb, ub, decisionVar, types, names);
lb = null;
ub = null;
names = null;
model.Update();

GRBExpr exp = model.GetObjective();
model.SetObjective(exp, GRB.MINIMIZE);

Console.WriteLine("done.\n");
Console.Write("Adding constraints...");

// Time periods
for (int t = 0; t < 12; t++)
{
    int leasePortsPeriod = t * m-params.numPorts * 2;
    int period = t * m-params.numPorts * m-params.numPorts;
    for (int i = 0; i < m-params.numPorts; i++)
    {
        // Setup constraints
        double[] constraint = new double[m-params.numDecisionVars];

        //Zeroise
        for (int k = 0; k < constraint.Length; k++)
        {
            constraint[k] = 0.0f;
        }

        double demand = 0.0;

        // Set Reposition Variable
        for (int j = 0; j < m-params.numPorts; j++)
        {
            if (i != j)
            {
                double transitDays = m-params.transitDays[i * numPorts + j +
period];
                int transitCeiling = (int)Math.Ceiling(transitDays / 30);
                int transitFloor = (int)Math.Floor(transitDays / 30);

                // Reposition Out
                constraint[i * numPorts + j + period] = -1;

                // Reposition In
                int offsetStartIdx = transitCeiling;
                int offsetEndIdx = transitFloor;
                int startIndex = offsetStartIdx > 1 ? offsetStartIdx : 1;
for (int i = startIdx; i < m_params.earlyRepoMonths; i++)
{
    // Early reposition out
    constraint[GetIdx(i, j, t, l)] = -1;

    // Early reposition in
    constraint[GetIdx(j, i, t - l, l)] = 1;
}

double repoStartPercent = (1.0 - (transitCeiling - transitDays / 30.0));
repoStartPercent;
repoStartPercent;

// Export Demand
double mean = m_params.demand_E[i * numPorts + j + period];
double Eij = m_params.coeffVariance > 0.0 ? Normal.Sample(m_rnd, mean, Math.Abs(mean * m_params.coeffVariance)) : mean;
Eij = Math.Round(Eij, 0);
if (Eij < 0) Eij = 0;

// Import
double detentionDays = m_params.detentionDays[i * numPorts + j + period];
mean = GetImportVolume(i, j, t, transitDays, detentionDays);
double Eji = (m_params.coeffVariance > 0.0 ? Normal.Sample(m_rnd, mean, Math.Abs(mean * m_params.coeffVariance)) : mean);
Eji = Math.Round(Eji, 0);
if (Eji < 0) Eji = 0;

demand += Eij - Eji;
}
else
{
    // Hold for next months
    for (int l = 1; l < m_params.earlyRepoMonths; l++)
    {
        // stock for next months
        constraint[GetIdx(i, i, t, l)] = -1;

        // stock from previous months
        constraint[GetIdx(i, i, t - l, l)] = 1;
    }
}

// Set leasing variable
// Onhire
constraint[numPortPairsMonth * m_params.earlyRepoMonths + i + leasePortsPeriod] = 1;

// Offhire
constraint[numPortPairsMonth * m_params.earlyRepoMonths + numPorts + i + leasePortsPeriod] = -1;
// Make sure that the flow is zero for every month!
GRBLinExpr ex1 = new GRBLinExpr();
for (int p = 0; p < vars.Length; p++)
{
    if (constraint[p] != 0.0)
    {
        ex1.AddTerm(constraint[p], vars[p]);
        // Testing
        // if (s != "") s += ", ";
        // s += "[" + p + "]=" + Math.Round(constraint[p], 6);
    }
}

// Console.WriteLine(s +" >= " + demand);
// Constraint (1)
model.AddConstr(ex1, GRB.GREATER_EQUAL, demand, "C" + t *
m_params.numPorts + i);
model.AddConstr(ex1, GRB.LESS_EQUAL, demand + 1, "C" + t *
m_params.numPorts + i + 1);

GRBLinExpr ex2 = new GRBLinExpr();
// Constraint (2)
if (demand < 0)
{
    for (int l = 1; l < m_params.earlyRepoMonths; l++)
    {
        ex2.AddTerm(1, vars[GetIdx(i, i, t, l)]);
    }
    model.AddConstr(ex2, GRB.LESS_EQUAL, -1.0 * demand, "C" + t *
m_params.numPorts + i + 2);
}
else
{
    for (int l = 1; l < m_params.earlyRepoMonths; l++)
    {
        ex2.AddTerm(1, vars[GetIdx(i, i, t, l)]);
    }
    model.AddConstr(ex2, GRB.LESS_EQUAL, 0, "C" + t *
m_params.numPorts + i + 2);
}
}

Console.WriteLine("done.
");

// Solve
Console.WriteLine("Solving...");
model.Optimize();

int optimstatus = model.Get(GRB.IntAttr.Status);
String solveResultsType = "";
solveResultsType = GetSolveResultsType(optimstatus);

double objective = model.Get(GRB.DoubleAttr.ObjVal);
double stockPerMonthMean = 0.0;
double stockPerMonthStdDev = 0.0;

GetAverageStockPerMonth(ref vars, ref stockPerMonthMean, ref stockPerMonthStdDev);

int numTrials = m_params.numTrials;
double sum = 0;
timer.Stop();
Console.WriteLine("Solution({0}-{1}): {2} (time={3} min {4} s)", 0,
solveResultsType, objective,
Convert.ToInt32(timer.ElapsedMilliseconds / 60 / 1000),
Convert.ToInt32(timer.ElapsedMilliseconds / 1000 % 60)
);

m_results.elapsedTimes.Add((int)(timer.ElapsedMilliseconds / 1000.0));
timer.Reset();
timer.Start();

m_results.objectives.Add(objective);
m_results.stockPerMonthMean.Add(stockPerMonthMean);
m_results.stockPerMonthStdDev.Add(stockPerMonthStdDev);
sum += objective;

// Reset Model
if (numTrials > 1) model.Reset();

for (int s = 1; s < numTrials; s++)
{
    for (int t = 0; t < 12; t++)
    {
        int period = t * m_params.numPorts * m_params.numPorts;

        for (int i = 0; i < m_params.numPorts; i++)
        {
            double demand = 0.0;

            for (int j = 0; j < m_params.numPorts; j++)
            {
                if (i != j)
                {
                    double transitDays = m_params.transitDays[i * numPorts + j + period];

                    // Export Demand
                    double mean = m_params.demand_E[i * numPorts + j + period];
                    double Eij = m_params.coeffVariance > 0.0 ?
                        Normal.Sample(m_rnd, mean, Math.Abs(mean * m_params.coeffVariance)) :
                        mean;
                    Eij = Math.Round(Eij, 0);
                    if (Eij < 0) Eij = 0;

                    // Import
                }
            }
        }
    }
}
double detentionDays = m_params.detentionDays[i * numPorts + j + period];
mean = GetImportVolume(i, j, t, transitDays, detentionDays);

double Eji = (m_params.coeffVariance > 0.0 ? Normal.Sample(m_rnd, mean, Math.Abs(mean * m_params.coeffVariance)) : mean);
Eji = Math.Round(Eji, 0);
if (Eji < 0) Eji = 0;
demand += Eij - Eji;
}

GRBConstr[] constrs = model.GetConstrs();
// Constraint (1)
GRBConstr c0 = constrs[(i + numPorts * t) * 3];
c0.Set(GRB.DoubleAttr.RHS, demand);

if (demand < 0)
{
    c0.Set(GRB.DoubleAttr.RHS, -1.0 * demand);
}
else
{
    c0.Set(GRB.DoubleAttr.RHS, 0.0);
}

// Constraint (2)
c0 = constrs[(i + numPorts * t) * 3 + 1];
c0.Set(GRB.DoubleAttr.RHS, demand + 1);

if (demand < 0)
{
    c0.Set(GRB.DoubleAttr.RHS, -1.0 * demand);
}
else
{
    c0.Set(GRB.DoubleAttr.RHS, 0.0);
}

// Now solve the model
model.Optimize();
optimstatus = model.Get(GRB.IntAttr.Status);
solveResultsType = GetSolveResultsType(optimstatus);
objective = model.Get(GRB.DoubleAttr.ObjVal);
GetAverageStockPerMonth(ref vars, ref stockPerMonthMean, ref stockPerMonthStdDev);
sum += objective;
timer.Stop();
Console.WriteLine("Solution({0}-{1}): {2} (time={3} min {4} s)", s, solveResultsType, objective,
Convert.ToInt32(timer.ElapsedMilliseconds / 60 / 1000),
Convert.ToInt32(timer.ElapsedMilliseconds / 1000 % 60)
);

m_results.elapsedTimes.Add((int)(timer.ElapsedMilliseconds / 1000.0));
timer.Reset();
timer.Start();

m_results.objectives.Add(objective);
m_results.stockPerMonthMean.Add(stockPerMonthMean);
m_results.stockPerMonthStdDev.Add(stockPerMonthStdDev);
}
timer.Stop();
LogWriteLine("Expected Solution: "+ sum / numTrials);

// Save the results
{
    m_results.solveResults = solveResultsType;
    m_results.expectedObjective = sum / numTrials;

    m_results.repo_out = new double[numPortPairsMonth *
       m_params.earlyRepoMonths];
    m_results.cost_coeff = new double[numPortPairsMonth *
       m_params.earlyRepoMonths];
    m_results.lease = new double[numPorts * 12];
    m_results.offhire = new double[numPorts * 12];
    m_results.numPortPairsMonth = numPortPairsMonth;
    m_results.numPorts = numPorts;

    for (int l = 0; l < m_params.earlyRepoMonths; l++)
    {
        for (int i = 0; i < numPortPairsMonth; i++)
        {
            m_results.repo_out[i + m_params.numPortPairsMonth * l] = vars[i +
               m_params.numPortPairsMonth * l].Get(GRB.DoubleAttr.X); // resultsDescisionVar[i];
            m_results.cost_coeff[i + m_params.numPortPairsMonth * l] =
               decisionVar[i];
        }
    }

    for (int t = 0; t < 12; t++)
    {
        int leasePortsPeriod = t * m_params.numPorts * 2;

        for (int i = 0; i < numPorts; i++)
        {
            m_results.lease[i + t * numPorts] = vars[numPortPairsMonth *
               m_params.earlyRepoMonths + i + leasePortsPeriod].Get(GRB.DoubleAttr.X);

            m_results.offhire[i + t * numPorts] = vars[numPortPairsMonth *
               m_params.earlyRepoMonths + numPorts + i + leasePortsPeriod].Get(GRB.DoubleAttr.X);
        }
    }
}
model.Dispose();
env.Dispose();

private double GetImportVolume(int i, int j, int t, double transitDays, double detentionDays)
{

double mean = 0.0;

int transitCeiling = (int)Math.Ceiling((transitDays + detentionDays) / 30);
int transitFloor = (int)Math.Floor((transitDays + detentionDays) / 30);

int offsetStartIdx = transitCeiling;
int offsetEndIdx = transitFloor;
double kStart = 1.0;
double kEnd = 1.0;

if (detentionDays == 0) detentionDays = 6.0;
if (t - offsetStartIdx < 0)
{
    kStart = 1.0; // Growth trend for location
}
if (t - offsetEndIdx < 0)
{
    kEnd = 1.0; // Growth trend for location
}

double demandStartPercent = (1.0 - (transitCeiling - transitDays/30.0));
mean = m_params.demand_E[GetIdx(j, i, t - offsetStartIdx, 0)] * kStart * demandStartPercent;
mean += m_params.demand_E[GetIdx(j, i, t - offsetEndIdx, 0)] * kEnd * (1.0 - demandStartPercent);

return mean;

private String GetSolveResultsType(int optimstatus)
{
    return optimstatus == GRB.Status.OPTIMAL ? "OPTIMAL" : "INFEASIBLE";
}