8-2022

Optimal Global Supply Chain and Warehouse Planning under Uncertainty

Avnish Kishor Malde
amalde@clemson.edu

OPTIMAL GLOBAL SUPPLY CHAIN AND WAREHOUSE PLANNING UNDER UNCERTAINTY

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Industrial Engineering

---

by
Avnish Kishor Malde
August 2022

Accepted by:
Dr. Tuğçe Işık, Committee Chair
Dr. Mary E. Kurz
Dr. Scott J. Mason
Dr. Yongjia Song

Abstract

A manufacturing company's inbound supply chain consists of various processes such as procurement, consolidation, and warehousing. Each of these processes is the focus of a different chapter in this dissertation.

The manufacturer depends on its suppliers to provide the raw materials and parts required to manufacture a finished product. These suppliers can be located locally or overseas with respect to the manufacturer's geographic location. The ordering and transportation lead times are shorter if the supplier is located locally. Just In Time (JIT) or Just In Sequence (JIS) inventory management methods could be practiced by the manufacturer to procure the raw materials and parts from the local supplier and control the inventory levels in the warehouse. In contrast, the lead time for the orders placed with an overseas supplier is usually long because sea-freight is often used as a primary mode of transportation. Therefore, the orders for the raw materials and parts (henceforth, we collectively refer to raw material and part by part) procured from overseas suppliers are usually placed using forecasted order quantities. In Chapter 2, we study the procurement process to reduce the overall expected cost and determine the optimal order quantities as well as the mode of transportation for procurement under forecast and inventory uncertainty. We formulate a two-stage stochastic integer programming model and solve it using the progressive hedging algorithm, a scenario-based decomposition method.

Generally, the orders are placed with overseas suppliers using weekly or monthly forecasted demands, and the ordered part is delivered using sea-containers since sea-freight is the primary mode of transportation. However, the end manufacturing warehouse is usually designed to hold around one to two days of parts. To replenish the inventory levels, the manufacturer considered in this research unloads the sea-container that contains the part that needs to be restocked entirely. This may cause over-utilization of the manufacturer's warehouse if an entire week's supply of part is consolidated into a single sea-container. This problem is further aggravated if the manufacturer procures hundreds of different parts from overseas suppliers

and stores them in its warehouse. In Chapter 3, we study the time-series forecasting models that help predict the manufacturing company's daily demand quantities for parts with different characteristics. The manufacturer can use these forecasted daily demand quantities to consolidate the sea-containers instead of the weekly forecasted demand. In most cases, there is some discrepancy between the predicted and actual demands for parts, due to which the manufacturer can either have excess inventory or shortages. While excess inventory leads to higher inventory holding costs and warehouse utilization, shortages can result in substantially undesirable consequences, such as the total shutdown of production lines. Therefore, to avoid shortages, the manufacturer maintains predetermined safety stock levels of parts with the suppliers to fulfill the demands arising from shortages. We formulate a chance-constraint optimization model and solve it using the sample approximation approach to determine the daily safety stock levels at the supplier warehouse under forecast error uncertainty.

Once the orders are placed with the local and overseas suppliers, they are consolidated into trailers (for local suppliers) and sea-containers (for overseas suppliers). The consolidated trailers and sea-containers are then delivered to the manufacturing plant, where they are stored in the yard until they are called upon for unloading. A detention penalty is incurred on a daily basis for holding a trailer or sea-container. Consolidating orders from different suppliers helps maximize trailer and sea-container space utilization and reduce transportation costs. Therefore, every sea-container and trailer potentially holds a mixture of parts. When a manufacturer needs to replenish the stocks of a given part, the entire sea-container or trailer that contains the required part is unloaded. Thus, some parts that are not imminently needed for production are also unloaded and stored inside the manufacturing warehouse along with the required parts. In Chapter 4, we study a multi-objective optimization model to determine the sea-containers and trailers to be unloaded on a given day to replenish stock levels such that the detention penalties and the manufacturing warehouse utilization are minimized.

Once a sea-container or trailer is selected to replenish the warehouse inventory levels, its contents (i.e., pallets of parts) must be unloaded by the forklift operator and then processed by workers to update the stock levels and break down the pallets if needed. Finally, the unloaded and processed part is stored in the warehouse bins or shelves. In Chapter 5, we study the problem of determining the optimal team formation such that the total expected time required to unload, process, and store all the parts contained in the sea-containers and trailers selected for unloading on a given day is minimized.

Dedication

To Ma, Bapu, Deep, and Juhi.

Acknowledgments

First and foremost, I am incredibly thankful to my research advisor Dr. Tuğçe Işık for the invaluable guidance, continuous support, and tutelage during the course of my Ph.D. study. With her immense knowledge and experience, she provided me with insightful comments and suggestions at every stage of the research that helped me improve my research studies.

Next, I would like to thank the dissertation committee members, Drs. Mary E. Kurz, Scott J. Mason, and Yongjia Song. They provided me with constructive criticism that was beneficial in improving my overall research. Their invaluable advice was very influential in shaping my dissertation research as it helped me improve the research studies.

I would also like to thank BMW Manufacturing Company (BMW), located in Spartanburg, South Carolina, for funding the research and providing me with an opportunity to solve real-world business problems. I want to especially thank Guenter Thumser, Carrie Cline, and Art Gantt for their unwavering support and for providing me with all the necessary resources such as data and system access that helped me study various business processes and optimize them using my dissertation research. Apart from them, many other people from BMW have also helped me with the explaining the business processes and providing me the necessary resources, and I am thankful to all of them.

I want to acknowledge Clemson University's generous allotment of computing time on the Palmetto Cluster, which helped me run the experimental studies smoothly. Further, I would like to thank Grigori Yourganov, Asher Antao, and Linh Ngo for helping me whenever I had questions or problems related to running experiments on the Palmetto Cluster.

Lastly, I would like to thank my friends Jayesh, Yash, Mahek, Omkar, and Abhya for always being there for me when I needed them the most.

Table of Contents

Title Page . . . . **i**

Abstract . . . . **ii**

Dedication . . . . **iv**

Acknowledgments . . . . **v**

List of Tables . . . . **viii**

List of Figures . . . . **x**

1 Introduction **1**

2 Optimal Transportation Mode Selection and Capacity Allocation under Uncertainty **6**
2.1 Introduction 6
2.2 Literature review 8
2.3 Problem formulation 11
2.4 Solution methodology 18
2.5 Numerical study 26
2.6 Conclusions 36
2.7 Potential future research 36

3 Optimal Sea-container Consolidation under Uncertainty and Vendor Hub Inventory Control **38**
3.1 Introduction 38
3.2 Literature review 39
3.3 Problem description and solution approach 43
3.4 Time series forecasting study 45
3.5 Optimization problem formulation 56
3.6 Solution methodology and numerical study 58
3.7 Conclusions 69
3.8 Potential future research 70

4 Optimal Containers Selection with Potentially Conflicting Objectives **72**
4.1 Introduction 72
4.2 Literature review 74
4.3 Optimization model development 75
4.4 Solution strategies 79
4.5 Experimental setup 82
4.6 Data requirements 85
4.7 Results 86
4.8 Pilot study 89

4.9 Observations and conclusions . . . 92

5 Team Formation and Job Assignment to Improve Warehouse Operations . . . 94
5.1 Introduction . . . 94
5.2 Literature review . . . 95
5.3 Problem description and simulation model . . . 97
5.4 Experimental setup . . . 103
5.5 Results . . . 104
5.6 Conclusions . . . 108

6 Conclusions, Research Contributions and Future Research . . . 110

Bibliography . . . 114

List of Tables

2.1 Input parameters notation and description . . . 13
2.2 Decision variables notation and description . . . 14
2.3 Input parameters values for the experiments with LHS method . . . 27
2.4 $\mathbb{E}[\hat{v}_N]$ estimate using LHS . . . 28
2.5 σ of $\mathbb{E}[\hat{v}_N]$ estimate using LHS . . . 28
2.6 Average solution time and average sample generation time per replication using LHS (in seconds) . . . 28
2.7 Optimality gap using LHS . . . 29
2.8 Average number of iterations per replication for PHA using LHS . . . 29
2.9 Percentage convergence for PHA using LHS . . . 29
2.10 Input parameters values for the experiments with simple random sampling . . . 30
2.11 $\mathbb{E}[\hat{v}_N]$ estimate using simple random sampling . . . 30
2.12 σ of $\mathbb{E}[\hat{v}_N]$ estimate using simple random sampling . . . 30
2.13 Average solution time and average sample generation time per replication using simple random sampling . . . 30
2.14 Optimality gap using simple random sampling . . . 31
2.15 Input parameters values for the experiments ran to study the effects of different levels of forecast and inventory error uncertainties . . . 32
2.16 Forecast error uncertainty levels based on mean and standard deviation . . . 32
2.17 Inventory error uncertainty levels based on mean and standard deviation . . . 32
2.18 $\mathbb{E}[\hat{v}_N]$ estimate and σ of $\mathbb{E}[\hat{v}_N]$ estimate . . . 33
2.19 Optimality gap and average run time per replication . . . 33
2.20 Average number of iterations per replication and percentage convergence for B-PHA . . . 34
2.21 Value of stochastic solution . . . 35

3.1 Description of the abbreviations used in Tables 3.2 – 3.8 . . . 48
3.2 Average of daily MAPE values for individual models obtained using different schemes for high runner parts . . . 49
3.3 Average of daily MAPE values for individual models obtained using different schemes for low runner parts . . . 50
3.4 Average of daily MAPE values for ensemble models obtained using different schemes for high runner parts . . . 51
3.5 Average of daily MAPE values for ensemble models obtained using different schemes for low runner parts . . . 52
3.6 Average of daily MAPE values for individual models (using 89 Rolling - MAPE scheme) . . 53
3.7 Average daily MAPE values for Ensemble models (using 89 Rolling MAPE scheme) . . . 54
3.8 Average weekly MAPE values for WA SP and the current model used by the industrial partner 55
3.9 Certain input parameters notation and description . . . 56
3.10 Uncertain input parameters notation and description . . . 57
3.11 Decision variables notation and description . . . 57
3.12 Optimality gap for the high runner part with at least 94.5% confidence . . . 60

3.13 Optimality gap for the low runner part with at least 94.5% confidence 60
3.14 Average number of feasible solutions obtained each week for the high runner part 62
3.15 Average number of feasible solutions obtained each week for the low runner part 62
3.16 Weekly excess emergency stock at the supplier for the high runner part 64
3.17 Weekly excess emergency stock at the supplier for the low runner part 65

4.1 Notations used in multi-objective optimization model 77
4.2 Experimental models descriptions 85
4.3 Overall and one day ahead prediction accuracy of all models 87

5.1 Summary of the process time distributions 101
5.2 Input parameters notation and description 101
5.3 Decision variables notation and description 101
5.4 Currently practiced team formation 104
5.5 OptQuest solution for best team formation 104
5.6 Proposed solution for best team formation 105
5.7 Performance improvement 106
5.8 Average resource utilization using currently practised team formation 107
5.9 Average resource utilization using proposed solution 107
5.10 Resource utilization for proposed solution 108

List of Figures

1.1 Overall supply chain representation along with flow of goods . . . 2
1.2 Inbound supply chain representation along with transportation activities and flow of goods . 3

2.1 Overseas supply chain . . . 7

3.1 Splitting weekly order quantities into daily . . . 43
3.2 Ensemble forecasting model to predict daily demand values . . . 47
3.3 Average solution risk observed each week for the high runner part . . . 61
3.4 Average solution risk observed each week for the low runner part . . . 61
3.5 Number of feasible solutions obtained every week for the high runner part . . . 63
3.6 Number of feasible solutions obtained every week for the low runner part . . . 63
3.7 Weekly excess emergency stock at the supplier and shortages at manufacturer for the high runner part . . . 66
3.8 Weekly excess emergency stock at the supplier and shortages at manufacturer for the low runner part . . . 67
3.9 Comparing excess inventory of the high runner part at the end of each day . . . 68
3.10 Comparing excess inventory of the low runner part at the end of each day . . . 68

4.1 Warehouse and yard operations for warehouse stock replenishment . . . 73
4.2 NSGA-II flow-chart . . . 81
4.3 Non-dominated fronts . . . 81
4.4 Population size = 30 and max number of generation = 30 . . . 83
4.5 Population size = 60 and max number of generation = 30 . . . 83
4.6 Population size = 50 and max number of generation = 250 . . . 83
4.7 Pareto front generated using the ε-constraint method . . . 84
4.8 Uncapacitated ε-constraint method for sea-cans using maximum ε value . . . 88
4.9 Capacitated ε-constraint method for sea-cans using maximum ε value . . . 88
4.10 Uncapacitated ε-constraint method for sea-cans and trailers using maximum ε value . . . 88
4.11 Capacitated ε-constraint method for sea-cans and trailers using maximum ε value . . . 88
4.12 Uncapacitated ε-constraint method for sea-cans using minimum ε value . . . 88
4.13 Capacitated ε-constraint method for sea-cans using minimum ε value . . . 88
4.14 Uncapacitated ε-constraint method for sea-cans and trailers using minimum ε value . . . 88
4.15 Capacitated ε-constraint method for sea-cans and trailers using minimum ε value . . . 88
4.16 Uncapacitated Goal Programming for sea-cans . . . 89
4.17 Capacitated Goal Programming for sea-cans . . . 89
4.18 Uncapacitated Goal Programming for sea-cans and trailers . . . 89
4.19 Capacitated Goal Programming for sea-cans and trailers . . . 89
4.20 Decision flow of selected model . . . 90
4.21 Pilot study Day 2 to Day 6 . . . 91
4.22 Pilot study Day 7 to Day 11 . . . 91
4.23 Pilot study Day 12 to Day 16 . . . 91

4.24 Pilot study Day 17 to Day 21 91
4.25 Daily LU unloaded and total detention days achieved 92

5.1 Process flow chart 98
5.2 Queue representation of the process 99
5.3 Representation of the simulation model 100

Chapter 1

Introduction

The supply chain (SC) of a manufacturing company comprises several different parties including suppliers, transporters, warehouses, and consolidation centers. These parties perform different activities such as ordering, production, and transportation that help fulfill customer demand. Based on the demand or its forecast, the manufacturer places orders with its suppliers for the raw materials and parts required to manufacture a finished product. The transportation service provider then performs activities such as milk runs and consolidation to transport these raw materials and parts from the suppliers to the manufacturer. The warehousing and storage activities are performed at the manufacturer once these raw materials and parts are delivered to the manufacturer. The manufacturer then uses these raw materials and parts to produce the end product. The finished products are stored in the warehouses, from where they would be provided to customers either directly or through the network of distributors and retailers. The transportation service partners support the outbound distribution activities using various modes of transportation such as rail, road, air, and maritime. The supply chain from suppliers to the manufacturing plant is called the inbound supply chain. In contrast, the supply chain from the manufacturing plant to the customers is referred to as the outbound supply chain. An example of a manufacturing supply chain is provided in Figure 1.1. In this dissertation, we focus on optimizing the inbound supply chain operations of a manufacturing company. Therefore, we refer to the inbound supply chain as simply *supply chain* in the remainder of this dissertation.

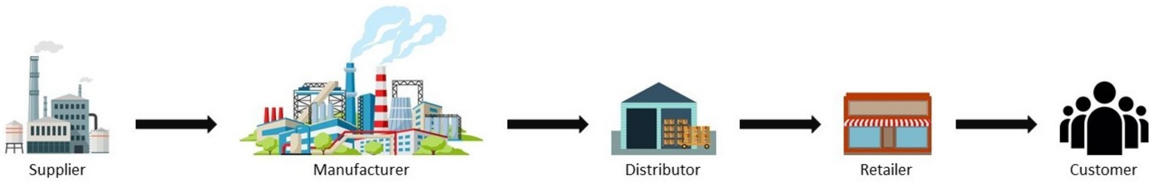


Figure 1.1: Overall supply chain representation along with flow of goods

The supply chain of a manufacturing company can span the globe with various suppliers located across different corners of the world. The manufacturers often order raw materials and parts using forecasted demand. The manufacturer can also practice the Just In Time (JIT) or Just In Sequence (JIS) procurement policies for some raw materials and parts. When using JIT and JIS procurement policies, the manufacturer places orders with the suppliers using deterministic demand. This dissertation only focuses on the raw materials and parts that the manufacturer procures using forecasted demand quantities. Further, we consider a global supply chain, wherein the suppliers are located both locally (i.e., located in the same country where the manufacturer is located) and across different parts of the globe.

In global supply chains, the ordering and transportation of raw materials and parts (henceforth, we refer to raw materials and parts by *materials*) are carried out under uncertainty due to errors in the forecast, changes in transportation mode availability and lead times, inventory levels, and supplier availability. These sources of uncertainty can cause disturbances in the daily functioning of the manufacturing company due to problems such as excessive inventory in the warehouse, stock-outs of parts, higher logistics and inventory-holding costs.

We consider a manufacturing company with the following supply chain configuration. The company has a single consolidation center located in Europe and multiple suppliers situated primarily across North America and Europe. The orders are placed with these suppliers using forecasted demand. Depending on the factors such as type of materials, location of the supplier, and transportation lead times, the orders are placed using daily or weekly forecasts. The supplier then prepares the order, and material is delivered either by using in-house transportation or 3PL (Third Party Logistics) service. These transportation service providers perform the milk-runs to collect the materials from various suppliers and then delivers them to either the European Consolidation center (EC) (if the supplier is located in the European region) or directly to the manufacturing plant (if the supplier is located in North America). If a given material is required urgently due to reasons such as low inventory levels or an unexpected surge in demand, then the material is procured using air-freight. Although air-freight is a costly option, it prevents the complete shutdown of the production lines caused by material unavailability. The pictorial representation of the supply chain configuration considered in this study is provided in Figure 1.2.

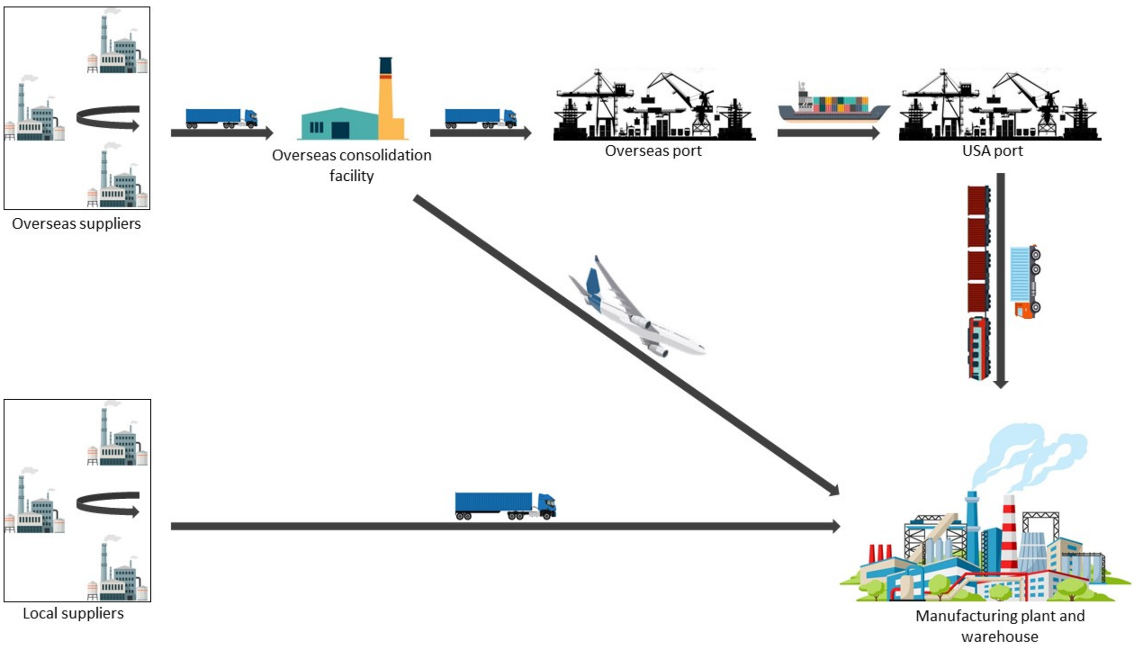


Figure 1.2: Inbound supply chain representation along with transportation activities and flow of goods

The materials procured from the European suppliers are delivered at EC via trailers at predetermined times. Upon arrival, all materials are unloaded from the trailer and consolidated into the sea-containers (sea-cans/cans) based on predefined loading rules. The EC does not have any inventory holding capacity as it is a cross-docking facility. Therefore, all unloaded materials from inbound trailers must be loaded into outbound sea-containers. However, EC does have a repackaging facility. It can repackage the materials and then load them into the sea-containers if required. Materials are repackaged depending on factors such as the type of packaging provided by the supplier and the total order quantity for a given week. These packed sea-containers are delivered to the port of origin (in Europe), which are then transported to the port of unloading (in the USA) using sea-freight. The sea-freight arrives once a week. The sea-containers are then delivered from the port of unloading to the manufacturing plant using a combination of rail and road transportation. Almost all sea-containers contain multiple materials.

Upon delivery to the plant, all sea-containers and trailers are stored at an inland port or yard until they are called for unloading into the warehouse. Detention penalties are incurred on a daily basis due to holding sea-containers and trailers for a longer duration than permitted by the sea-container and trailer contracting company. For each material with an inventory level below a specified threshold value, the sea-containers or trailers containing the material are unloaded into the warehouse. If multiple sea-containers or trailers containing the required material are available, the tie is broken by selecting the one that minimizes the detention penalties incurred.

Inside the warehouse, workers work in teams to complete the replenishment process. When the sea-container or trailer needs to be unloaded, it is assigned to one of the teams. A team comprises of the unload-forklift operator, receivers, and breakdowners. The unload-forklift unloads all the material; the receiver scans all the material and updates the warehouse stock in ERP (Enterprise Resource Planning) system. Some of the unloaded material is stored in pallets, while the rest needs to be broken down from pallets and stored in individual boxes. Therefore, breakdowners are required to break down the pallets of these materials. There is a limited staging area where the process of scanning and breaking down can be done. After all these processes are completed, the transporter-forklifts clear the staging area by storing all the material in the warehouse aisles.

In this dissertation, we study the problem of optimizing a global supply chain under uncertainty such that: (i.) the overall cost of procurement, inventory holding, and back-orders is minimized, (ii.) the number of remnant materials in the warehouse is minimized on a daily basis, (iii.) the detention penalties of sea-containers and trailers are minimized, and (iv.) the unloading times of sea-containers and trailers are minimized. All numerical studies in this dissertation have been performed using the data obtained from our industrial partner. Although, this dissertation considers a manufacturing plant that is located in the USA with its suppliers located across North America and Europe, all problems we study we study can be generalized to any company with a global supply chain with various parties at different geographical locations and lead times.

The remainder of this dissertation is organized as follows. In chapter 2, we study the optimal order quantities for globally sourced materials using each available transportation mode with different lead times, under forecast and inventory error uncertainty. Our goal is to minimize the expected long-term cost of order procurement, inventory holding, and back-order costs. We formulate this problem as a two-stage stochastic integer program and solve it using Sample Average Approximation (SAA) method and Progressive Hedging Algorithm (PHA).

In chapter 3, we study the problem of determining the sea-container consolidation quantities using daily demand and emergency order stock levels at the supplier warehouse under forecast errors. We assume that currently, the orders are placed with the suppliers using weekly forecasted demand by the manufacturer.

However, the final production warehouse is designed to hold less than a week's materials supply. Therefore, the weekly order quantities must be disintegrated into daily quantities, and the sea-containers must be packed to minimize daily warehouse utilization. We build and study different time-series models to forecast the daily demands of various parts. Further, we use the chance constraint optimization model to determine the supplier inventory levels for fulfilling emergency orders arising from shortages under forecast error uncertainty.

In chapter 4, we consider a warehouse with multi-product inventory, where the materials are continuously consumed based on the line-side demand. The inventory level is replenished using materials available in trailers and sea-containers stored in the inland port or yard of the manufacturing company. These sea-containers and trailers arrive over time and a daily detention penalty is incurred for holding them in the inland port or the yard. We study the problem of optimally selecting the sea-container and trailers to reduce the number of remnant materials in the warehouse and minimize the detention penalties. We formulate this problem as a multi-objective integer program. The multi-objective optimization problem is solved using goal programming and the ε-constraint method. A detailed discussion on the experimental setup and the numerical results is also presented.

In chapter 5, we study the problem of optimizing the container unloading and warehouse replenishment process. To solve this problem, we determine the optimal team formation, buffer allocation, and job assignment to teams of servers. The teams work at two tandem stations, each with several finite-buffered parallel queues. The objective of the problem is to minimize the total time required to process all jobs. There are operational constraints such as the number of teams that can be formed, the number of servers in each team, and the number of buffer spaces allotted to each team. Additionally, precedence constraints on the order of processing each job also need to be considered. We solve this problem using a simulation optimization method.

Chapter 2

Optimal Transportation Mode Selection and Capacity Allocation under Uncertainty

2.1 Introduction

A manufacturer requires several different raw materials and parts to manufacture a finished product. These raw materials can be procured from the suppliers located locally (i.e., in the same geographical region as the manufacturer) or overseas. In this chapter, we will focus on the procurement of parts from suppliers located overseas. For the raw materials and parts procured from an overseas supplier, the manufacturer places orders based on the forecasted quantities. Sea-freight is the primary mode of transportation for procuring materials from overseas suppliers because of its cost-effectiveness. Although the sea-freight option is cost-effective, the transportation lead time for the maritime transportation is high. Situations in which the manufacturer requires the material as soon as possible can arise. In case of emergencies, air-freight can also be used to help with the faster movement of materials as it has lower transportation lead times.

In this chapter, we consider a manufacturing plant located in the USA. We focus only on the overseas supply chain of the manufacturer, as shown in Figure 2.1. The manufacturer procures the raw materials and parts from several suppliers located in Europe. The orders are placed with these suppliers using weekly forecasted demand. The supplier prepares the order, the ground transportation service provider collects these orders from various suppliers and delivers them to the European consolidation center. At the consolidation center, the materials delivered by the ground transportation services are consolidated into the sea-containers. These sea-containers are then sent to the European port and transported to the port in the USA. The sea-freight orders arrive once a week. Once these sea-containers are delivered to the port in the USA, a combination of rail and road transport is used to deliver them to the manufacturing plant. The manufacturer prefers the maritime as a primary mode of transportation because of cost-effective reasons.

All materials procured from the overseas region are stored in the warehouse. Occasionally the warehouse inventory levels shown in the warehouse management system (WMS) are not accurate due to reasons such as misplacement of material within the warehouse, quality issues, and over-usage. If a given material is required urgently due to reasons such as low inventory levels or an unexpected surge in demand, then the parts are procured using air-freight. Although air-freight is a costly option, it prevents the complete shutdown of the production lines.

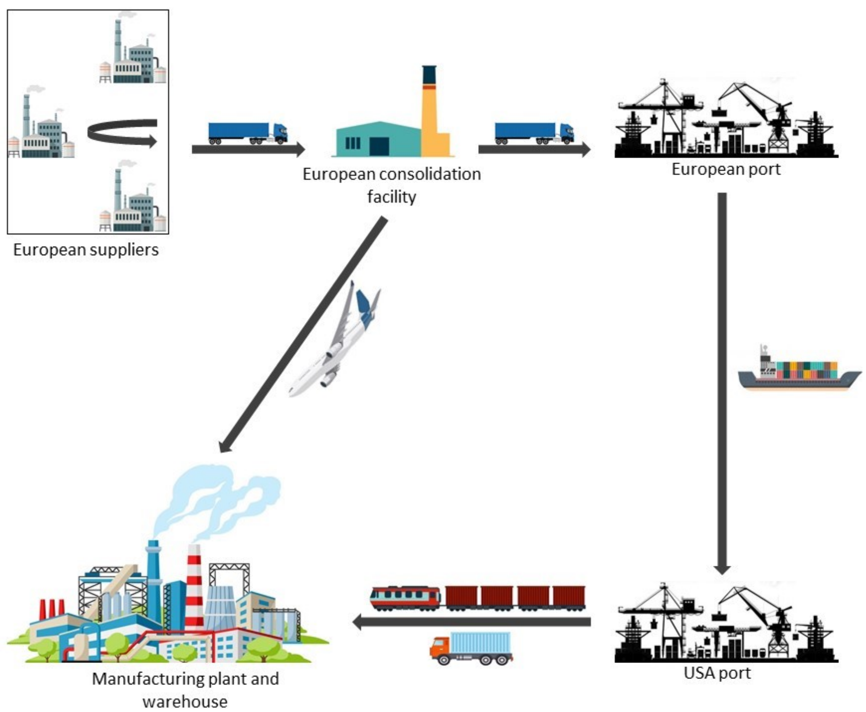


Figure 2.1: Overseas supply chain

In this chapter, we study the problem of optimal transportation mode selection and capacity allocation for procuring materials from overseas suppliers, considering forecast error and inventory level uncertainty. Although the manufacturer procures multiple products from overseas suppliers, we study a single-product problem because there are no capacity constraints on the transportation modes considered in this problem. Additionally, solving a single-product problem provides a natural decomposition for the multi-product problem and reduces the overall time required to generate solutions. We formulate the problem as a two-stage stochastic integer program. We use Sample Average Approximation (SAA) method to approximate the stochastic program and solve it using the Progressive Hedging Algorithm (PHA). Using this formulation, we study the impact of the following factors on the quality of solution and computational time required to solve the problem:

(i) Sampling methods such as simple random sampling and stratified sampling.

(ii) Solution methodology such as using exact methods versus decomposition based algorithms to solve an extensive form of the stochastic programming problem.

(iii) Different levels of the forecast error and inventory error uncertainty.

The remainder of this chapter is organized as follows. The next section provides a brief literature review on order placement and transportation mode selection under uncertainty. In Section 2.3, the problem is described and formulated. In Section 2.4, the solution methodology for our problem is discussed. In Section 2.5, the details for the experimental setup and results of the numerical study are presented using the synthetic data sets modeled utilizing the industrial data. Concluding remarks are provided in Section 2.6. Finally, in Section 2.7, we discuss a potential future research direction.

2.2 Literature review

In this problem, we focus on minimizing the long-run expected procurement, inventory holding, and backorder costs for a manufacturing company that uses multiple transportation modes for inventory replenishment. The orders can be split across multiple available transportation modes, i.e., modal splitting is allowed. Specifically, we study an integrated inventory and transportation model while allowing modal splitting. Further, the strategic-level supply chain decisions such as supplier selection and cross-dock/consolidation facility selection are already made and cannot be changed at the stage we aim to model. Therefore, the problem we formulate optimizes tactical and operational level decisions under uncertainty.

Engebrethsen and Dauzère-Pérès (2019) provides an extensive review of the inventory replenishment models using multiple transportation modes. According to Engebrethsen and Dauzère-Pérès (2019), the research in the area of modal splitting is underdeveloped. The literature related to integrated transportation-inventory models was studied exhaustively by Mosca et al. (2019). From both Engebrethsen and Dauzère-Pérès (2019) and Mosca et al. (2019), we observe that the studies considering stochastic demand are cited, but studies that model inventory errors as part of the uncertainty are not noticed. A multi-product multi-period global supply chain planning problem for production, distribution, and inventory decisions under demand and freight rate uncertainty was studied by You et al. (2009). You et al. (2009) formulated this problem using a two-stage stochastic linear programming approach. To solve the two-stage stochastic linear programming model You et al. (2009) used the multi-cut L-shaped method. Bidhandi and Yusuff (2011) studied a supply chain planning problem with uncertainty in operational costs, customer demand, and the capacities of the facilities. In the first-stage, strategic decisions corresponding to the supply chain configuration are made, and in the second-stage, tactical decisions such as processing and transportation of products from suppliers to the customers are made to solve the problem in Bidhandi and Yusuff (2011) optimally. Tolooie et al. (2020) formulate a two-stage stochastic mixed-integer programming model to study the supply chain network design under random demand and facility disruptions. In the first-stage, decisions on the selection of suppliers and facilities to be established are made, while in the second-stage, once the uncertainties are realized, how the customer demand should be fulfilled using the available facilities selected in the first stage is decided. The problem we study in this chapter is different since we decide the order quantities using each available mode of transportation under forecast and inventory error uncertainty.

A transportation mode selection problem for a global supply chain with supply chain node and transportation link uncertainty is studied by Fan et al. (2017). The authors formulated a two-stage stochastic programming model to select the transportation modes for various products to minimize the overall supply chain cost. Three modes of transportation, with high, medium, and low speed, are considered in this problem. In contrast to the Fan et al. (2017) wherein the supply chain planning using various transportation modes under disruption risk is studied, we study the problem of supply chain planning using various transportation modes under operational risk.

A two-stage stochastic programming problem is studied by Dillon et al. (2017) to minimize the ordering cost and the expected holding, shortage, and outdate cost for a blood supply chain. Dillon et al. (2017) consider the demand as a stochastic parameter, and in the first-stage, the decision on the replenishment policy (i.e., frequency of reviews (R) and target inventory levels for each blood type (S)) is made. In the second-stage, the decisions related to the daily operation of the system, such as the order quantity for each blood type and on-hand inventory, shortages, and obsolete inventory for each blood type in each period, are made once the demand is realized. Dehghani et al. (2021) also formulated a two-stage stochastic programming problem under demand uncertainty for a multi-period blood supply chain problem. However, in their paper, the order and transshipment quantities for the current period, along with the target inventory level (S), are determined as first-stage decisions, whereas the order quantities for the future period are determined as the second-stage decisions. Their objective is to minimize the order and transshipment costs for the current period (t=1), the expected holding, outdating, and shortage cost for all periods, and the expected order costs for future periods ($t \geq 2$). A two-stage stochastic programming problem with multiple objectives for a blood supply chain under demand and supply uncertainty was studied by Hamdan and Diabat (2019). They considered three minimization objectives: the total cost in the system, the delivery time, and the number of outdated units. The problem was solved using the ε-constraint method.

Perishable supply chains have been studied extensively in the literature. The problems related to perishable supply chains are of interest to us since they model inventory loss considering the deterioration rate function. A single-product perishable inventory control problem to determine the ordering quantities in different periods was studied by Pauls-Worm et al. (2016). They considered the demand to be non-stationary, and the objective of the problem was to minimize the fixed inventory replenishment cost, variable inventory replenishment cost, and the expected cost of inventory holding and wastage. Lin and Wang (2018) studied a problem to determine facility locations and inventory policy for each facility under demand uncertainty. A perishable supply chain problem under demand and supply uncertainty is studied using a two-stage stochastic programming model in Azadi et al. (2019). In the first-stage, decisions on supplier selection and pricing are made, while in the second-stage, inventory replenishment decisions are made. The above-mentioned perishable supply chain studies do not consider the inventory deterioration rate as a random parameter.

Transportation and supply chain related problems are also studied exhaustively in the context of humanitarian-aid logistics. Barbarosoğlu and Arda (2004) studied a multi-commodity, multi-modal transportation planning problem for a disaster relief operation using a two-stage stochastic programming model. The supply and demand of the commodities and the capacity of each transportation mode were considered to be uncertain. Afshar and Haghani (2012) studied a deterministic facility location and vehicle routing problem to fairly distribute the relief commodities across all the demand points using a mixed-integer programming formulation. A facility selection and inventory planning problem under demand and transportation link uncertainty was studied by Rawls and Turnquist (2012). They formulated a two-stage stochastic programming model with a reliability constraint to ensure that the demand is met in $\alpha\%$ of scenarios. Salas et al. (2012) studied a perishable inventory ordering problem under demand uncertainty for hurricane response. An optimal order quantity is determined in the first-stage using a two-stage stochastic programming model in Salas et al. (2012).

We consider random fluctuations in the on-hand inventory at the beginning of each period that are modeled by random error terms. Such errors can also be accounted for in the problems mentioned earlier, such as the perishable supply chains, blood supply chains, and humanitarian aid logistics. However, to the best of our knowledge, no study in the existing literature has accounted for the randomness in the inventory errors for the ordering and transportation mode selection problem. There can be errors in the tracked levels of on-hand inventory due to substitutions or misplacement of items, resulting in false loss of material. Moreover, a volatile demand does not necessarily mean highly uncertain demand. Generally, in industrial practice, the demand is forecasted using the ensemble modeling approach, i.e., the forecast is generated using various time series models. Therefore, instead of treating demand as a stochastic parameter, we consider the errors in the forecasted demand to be uncertain. In our problem, the cost of expediting an order and using an expensive but fast mode of transportation such as air-freight can be reduced when the randomness of the inventory errors are also accounted for during order placement.

2.3 Problem formulation

For the supply chain under consideration, sea-freight is the primary mode of transportation for procuring materials. The air-freight is used to fulfill the emergency orders arising due to low inventory levels in the manufacturing plant or sudden surge in demand for the material. Both transportation modes are considered to be available all the time. Further, both transportation modes can be used without any capacity constraints since any number of required transportation flights can be booked, and any number of required sea-containers can be booked.

We assume that the order procurement lead time using sea-freight is seven weeks. In contrast, the procurement lead times for express and slow air-freight are one week and two weeks respectively. The sea-freight order procurement lead time can be further broken down as follows: two weeks for delivering the order from supplier to consolidation center in Europe, one week for delivering the containers from consolidation center to the port in Europe, twelve days on the sea (European port to US port), four days on the rail (US port to the manufacturing plant) and twelve days of safety lead time (i.e., sea-cans containing these parts need to be at the manufacturing plant twelve days prior to the required week).

In our formulation, the current week is denoted as *week 0*. We assume that the *week i* order has been placed to meet the demand arising in *week i*. For instance, when a new order is placed for sea-freight, it is intended to meet the demand in *week 7*. When a new order is placed for material, there is already inventory of that material across different locations in the supply-chain which corresponds to the following quantities:

- The orders corresponding to week 0 (i.e., current week) and week 1 (i.e., next week) of the same material are already at the plant.
- Week 2 order is at the US port.
- Week 3 order is mid-way in the sea.
- Week 4 order is being shipped from the consolidation center to the port in Europe.
- Week 5 order is either being prepared at the consolidation center or on its way to EC.
- Week 6 order is either in transit to the consolidation center or being prepared by the supplier.

The demand for the current week is known accurately, and for the remaining weeks, the forecasted demand is available. The forecast values corresponding to the near future weekly demands are more accurate than those for far out in the future. Since the demand for the upcoming weeks is uncertain, there is a possibility for excess inventory or shortage of the required material if only sea-freight is used to procure the material. Further, the excess inventory at the beginning of each period (week) is uncertain due to reasons such as inventory misplacement, quality issues, and over-usage. Due to these reasons, there can be either excess inventory leftover or a shortage of the material. The resultant shortages can be met, and excess inventory reduced by ordering the material one or two weeks ahead of actual requirement (i.e., when the demand becomes less uncertain) using either slow or express air-freight.

Given the stochastic nature of the problem we described, we model our problem using a two-stage stochastic integer program to optimize the allocation of orders to different transportation options. Since the sea-freight is the primary mode of transportation for procuring the material and has the largest lead time, the first-stage decision is determines the order quantity to be procured using sea-freight as part of the next weekly order. The recourse decisions are for the orders that need to be procured using either slow or express air-freight to meet the excess demand arising due to uncertainty in the system. A detailed description of our two-stage stochastic integer problem is as follows.

Table 2.1: Input parameters notation and description

Certain input parameters	
D_0	Current week's demand
F_i	Forecasted order quantity for week i ($where\ i = 1,\ 2,\ 3,\ 4,\ 5,\ 6\ and\ 7$)
I_c	Current total inventory on hand (week 0 and week 1)
I_0	Excess inventory at the end of the week 0
S_0	Shortage at the end of the week 0
$Q_{i_{sea}}$	Quantity of parts ordered and transported via sea-freight for week i ($where\ i = 2,\ 3,\ 4,\ 5\ and\ 6$)
$Q_{0_{airexp}}$	Quantity of parts ordered and transported via express air-freight for week 0
$Q_{i_{air_{slow}}}$	Quantity of parts ordered and transported via slow air-freight for week i ($where\ i = 0\ and\ 1$)
c_{sea}	Per unit cost of shipping via sea-freight
c_{truck}	Per trip cost of hiring a delivery truck to transport the parts from US port to the plant
$c_{air_{express}}$	Per unit cost of shipping via express air-freight
$c_{air_{slow}}$	Per unit cost of shipping via slow air-freight
h	Per unit weekly holding cost
b	Per unit weekly back-order cost
p_k	Probability of scenario k
T	Total number of scenarios
Uncertain input parameters	
ξ_{i_δ}	Forecast error for week i ($where\ i = 1,\ 2,\ 3,\ 4,\ 5,\ 6\ and\ 7$)
ξ_{i_ι}	Inventory error for week i ($where\ i = 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6\ and\ 7$)
$\xi_\mathbf{k}$	A vector containing values of uncertain parameters for particular realization of scenario 'k' (where, $\xi_\mathbf{k}$ = $[\xi^k_{1_\delta},\ \xi^k_{2_\delta},\ \ldots, \xi^k_{7_\delta},\ \xi^k_{0_\iota},\ \xi^k_{1_\iota},\ \ldots,\ \xi^k_{7_\iota}]$)

Table 2.2: Decision variables notation and description

	First stage decision variables
$q_{7_{sea}}$	Quantity of parts to be ordered and transported via sea-freight for week 7
	Second stage decision variables
$q_{i_{sea}}$	Quantity of parts to be ordered and transported via sea-freight for week i ($where\ i = 8\ and\ 9$)
$q_{i_{air_{express}}}$	Quantity of parts to be ordered and transported via express air-freight for week i ($where\ i = 1,\ 2,\ 3,\ 4,\ 5,\ 6\ and\ 7$)
$q_{i_{air_{slow}}}$	Quantity of parts to be ordered and transported via slow air-freight for week i ($where\ i = 2,\ 3,\ 4,\ 5,\ 6\ and\ 7$)
x_i	Binary variable, equals to 1 if the quantity of parts to be delivered at port for week i + 2 is more than zero ($where\ i = 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6\ and\ 7$)
z_i	Binary variable, equals to 1 if the demand for the week i is more than on-hand inventory including containers in yard and incoming air-freights. ($where\ i = 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6\ and\ 7$)
w_i	Binary variable, equals to 1 if a trucking trip is made for delivery from US port to the plant, i.e. if both '$x_i = 1$' and '$z_i = 1$'. ($where\ i = 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6\ and\ 7$)
I_i	Quantity of excess inventory at the end of the week i ($where\ i = 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6\ and\ 7$)
S_i	Quantity of shortages at the end of the week i ($where\ i = 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6\ and\ 7$)

Two-stage Stochastic Programming Model

$$\text{minimize:} \quad c_{sea} * q_{7_{sea}} + E[cost_2(q_{7_{sea}},\ \xi)] \tag{2.1}$$

Subject to:

$$q_{7_{sea}} \in \mathbb{Z}_{\geq 0} \tag{2.2}$$

We define $cost_2(q_{7_{sea}},\ \xi)$ as follows:

$$cost_2(q_{7_{sea}},\ \xi) := \text{minimize } c_{sea} * \sum_{i=8}^{9} q_{i_{sea}} + c_{truck} * \sum_{i=0}^{7} w_i \\ + c_{air_{exp}} * \sum_{i=1}^{7} q_{i_{air_{exp}}} + c_{air_{slow}} * \sum_{i=2}^{7} q_{i_{air_{slow}}} + h * \sum_{i=0}^{7} I_i + b * \sum_{i=0}^{7} S_i. \quad (2.3)$$

Subject to:

1. Weekly demand must be met in all weeks:

$$\begin{aligned} I_c + \xi_{0_\iota} + Q_{2_{sea}} + Q_{0_{air_{exp}}} + Q_{0_{air_{slow}}} - I_0 + S_0 &= D_0 \\ I_0 + \xi_{1_\iota} + Q_{3_{sea}} + q_{1_{air_{exp}}} + Q_{1_{air_{slow}}} - I_1 + S_1 &= F_1 + \xi_{1_\delta} + S_0 \\ I_{i-1} + \xi_{i_\iota} + Q_{(i+2)_{sea}} + q_{i_{air_{exp}}} + q_{i_{air_{slow}}} - I_i + S_i &= F_i + \xi_{i_\delta} + S_{i-1} \ldots i \in \{2,\ 3,\ \&\ 4\} \\ I_{j-1} + \xi_{j_\iota} + q_{(j+2)_{sea}} + q_{j_{air_{exp}}} + q_{j_{air_{slow}}} - I_j + S_j &= F_j + \xi_{j_\delta} + S_{j-1} \ldots j \in \{5,\ 6\ \&\ 7\} \end{aligned} \quad (2.4)$$

2. '$x_i = 1$' if the quantity of parts received for the week $i+2$ is greater than zero at US port:

$$\begin{aligned} Q_{(i+2)_{sea}} - (M+\varepsilon) * x_i &\le 1 - \varepsilon \ldots i \in \{0,\ 1, \ldots,\ 4\} \\ q_{(j+2)_{sea}} - (M+\varepsilon) * x_j &\le 1 - \varepsilon \ldots j \in \{5,\ 6\ \&\ 7\} \end{aligned} \quad (2.5)$$

3. '$z_i = 1$' if the demand for the week i is greater than on hand inventory comprising of warehouse and inland port:

$$\begin{aligned} I_c + \xi_{0_\iota} + Q_{0_{air_{exp}}} + Q_{0_{air_{slow}}} + (M+\varepsilon) * z_0 &\ge (D_0 - 1) + \varepsilon \\ I_0 + \xi_{1_\iota} + q_{1_{air_{exp}}} + Q_{1_{air_{slow}}} + (M+\varepsilon) * z_1 &\ge (F_1 + \xi_{1_\delta} + S_0 - 1) + \varepsilon \\ I_{i-1} + \xi_{i_\iota} + q_{i_{air_{exp}}} + q_{i_{air_{slow}}} + (M+\varepsilon) * z_i &\ge (F_i + \xi_{i_\delta} + S_{i-1} - 1) + \varepsilon \end{aligned} \quad (2.6)$$

$$\ldots i \in \{2,\ 3, \ldots,\ 7\}$$

4. '$w_i = 1$' if a trucking trip is made for delivery from US port to the plant, i.e. if both '$x_i = 1$' and '$z_i = 1$':

$$\left.\begin{aligned} w_i &\leq x_i \\ w_i &\leq z_i \\ w_i &\geq x_i + z_i - 1 \end{aligned}\right\} \ldots i \in \{0,\ 1, \ldots,\ 7\} \tag{2.7}$$

5. Non-negative integer variables:

$$\begin{aligned} q_{i_{sea}} &\in \mathbb{Z}_{\geq 0} \quad \ldots i \in \{8,\ 9\} \\ q_{i_{air_{slow}}} &\in \mathbb{Z}_{\geq 0} \quad \ldots i \in \{2,\ 3, \ldots,\ 7\} \\ q_{i_{airexp}} &\in \mathbb{Z}_{\geq 0} \quad \ldots i \in \{1,\ 2, \ldots,\ 7\} \\ I_i &\in \mathbb{Z}_{\geq 0} \quad \ldots i \in \{0,\ 1, \ldots,\ 7\} \\ S_i &\in \mathbb{Z}_{\geq 0} \quad \ldots i \in \{0,\ 1, \ldots,\ 7\} \end{aligned} \tag{2.8}$$

The stochastic programming problem given by (2.1) – (2.8) is an infinite-dimensional optimization problem if the uncertain parameters belong to a continuous probability distribution. In practice, such infinite-dimensional problems are approximated by considering only finitely many realizations (T) of a random vector ξ, such that the sum of probabilities of all considered scenarios is one. We denote p_k as the probability with which a scenario k is realized. Using this information, we rewrite the problem given by (2.1) – (2.8) in its deterministic equivalent form called extensive form as follows.

Extensive Form of Two-stage Stochastic Program

$$\begin{aligned} \text{minimize:} \quad & c_{sea} * q_{7_{sea}} + \sum_{k=1}^{T} p_k * [c_{sea} * \sum_{i=8}^{9} q^k_{i_{sea}} + c_{truck} * \sum_{i=0}^{7} w^k_i \\ & + c_{air_{exp}} * \sum_{i=1}^{7} q^k_{i_{airexp}} + c_{air_{slow}} * \sum_{i=2}^{7} q^k_{i_{air_{slow}}} + h * \sum_{i=0}^{7} I^k_i + b * \sum_{i=0}^{7} S^k_i] \end{aligned} \tag{2.9}$$

Subject to:

1. Weekly demand must be met in all weeks:

$$
\begin{aligned}
I_c + \xi^k_{0_\iota} + Q_{2_{sea}} + Q_{0_{airexp}} + Q_{0_{air_{slow}}} - I^k_0 + S^k_0 = D_0 \ldots \forall k \\
I^k_0 + \xi^k_{1_\iota} + Q_{3_{sea}} + q^k_{1_{airexp}} + Q_{1_{air_{slow}}} - I^k_1 + S^k_1 = F_1 + \xi^k_{1_\delta} + S^k_0 \ldots \forall k \\
I^k_{i-1} + \xi^k_{i_\iota} + Q_{(i+2)_{sea}} + q^k_{i_{airexp}} + q^k_{i_{air_{slow}}} - I^k_i + S^k_i = F_i + \xi^k_{i_\delta} + S^k_{i-1} \ldots \forall k \ \& \ i \in \{2,\ 3,\ 4\} \\
I^k_4 + \xi^k_{5_\iota} + q_{7_{sea}} + q^k_{5_{airexp}} + q^k_{5_{air_{slow}}} - I^k_5 + S^k_5 = F_5 + \xi^k_{5_\delta} + S^k_4 \ldots \forall k \\
I^k_{j-1} + \xi^k_{j_\iota} + q^k_{(j+2)_{sea}} + q^k_{j_{airexp}} + q^k_{j_{air_{slow}}} - I^k_j + S^k_j = F_j + \xi^k_{j_\delta} + S^k_{j-1} \ldots \forall k \ \& \ j \in \{6,\ 7\}
\end{aligned}
\tag{2.10}
$$

2. '$x^k_i = 1$' if the quantity of parts received for the week (i + 2) is greater than zero at US port:

$$
\begin{aligned}
Q_{(i+2)_{sea}} - (M+\varepsilon) * x^k_i \leq 1 - \varepsilon \ldots \forall\, k \ \& \ i \in \{0,\ 1, \ldots,\ 4\} \\
q_{7_{sea}} - (M+\varepsilon) * x^k_5 \leq 1 - \varepsilon \ldots \forall\, k \\
q^k_{(j+2)_{sea}} - (M+\varepsilon) * x^k_j \leq 1 - \varepsilon \ldots \forall\, k \ \& \ j \in \{6,\ 7\}
\end{aligned}
\tag{2.11}
$$

3. '$z^k_i = 1$' if the demand for the week i is greater than on hand inventory comprising of warehouse and inland port:

$$
\begin{aligned}
I_c + \xi^k_{0_\iota} + Q_{0_{airexp}} + Q_{0_{air_{slow}}} + (M+\varepsilon) * z^k_0 \geq (D_0 - 1) + \varepsilon \ldots \forall\, k \\
I^k_0 + \xi^k_{1_\iota} + q^k_{1_{airexp}} + Q_{1_{air_{slow}}} + (M+\varepsilon) * z^k_1 \geq (F_1 + \xi^k_{1_\delta} + S^k_0 - 1) + \varepsilon \ldots \forall\, k \\
I^k_{i-1} + \xi^k_{i_\iota} + q^k_{i_{airexp}} + q^k_{i_{air_{slow}}} + (M+\varepsilon) * z^k_i \geq (F_i + \xi^k_{i_\delta} + S^k_{i-1} - 1) + \varepsilon \ldots \forall\, k \ \& \ i \in \{2,\ 3, \ldots,\ 7\}
\end{aligned}
\tag{2.12}
$$

4. '$w^k_i = 1$' if a trucking trip is made for delivery from US port to the plant, i.e. if both '$x^k_i = 1$' and '$z^k_i = 1$':

$$
\left.
\begin{aligned}
w^k_i \leq x^k_i \\
w^k_i \leq z^k_i \\
w^k_i \geq x^k_i + z^k_i - 1
\end{aligned}
\right\} \ldots \forall\, k \ \& \ i \in \{0,\ 1, \ldots,\ 7\}
\tag{2.13}
$$

5. Non-negative integer variables:

$$\begin{aligned}
q_{7_{sea}} &\in \mathbb{Z}_{\geq 0} \\
q^k_{i_{sea}} &\in \mathbb{Z}_{\geq 0} \quad \ldots \forall k \;\&\; i \in \{8,\ 9\} \\
q^k_{i_{air_{slow}}} &\in \mathbb{Z}_{\geq 0} \quad \ldots \forall k \;\&\; i \in \{2,\ 3, \ldots,\ 7\} \\
q^k_{i_{air_{exp}}} &\in \mathbb{Z}_{\geq 0} \quad \ldots \forall k \;\&\; i \in \{1,\ 2, \ldots,\ 7\} \\
I^k_i &\in \mathbb{Z}_{\geq 0} \quad \ldots \forall k \;\&\; i \in \{0,\ 1, \ldots,\ 7\} \\
S^k_i &\in \mathbb{Z}_{\geq 0} \quad \ldots \forall k \;\&\; i \in \{0,\ 1, \ldots,\ 7\}
\end{aligned} \tag{2.14}$$

2.4 Solution methodology

In this section, we describe the components of our solution methodology. To approximate and solve the two-stage stochastic programming problem given in equations (2.1) – (2.8), we use Sample Average Approximation (SAA) method. In each iteration of SAA, we sample T realizations of each stochastic input parameter, i.e., we consider T scenarios. The resulting SAA problem is then given as follows.

SAA problem

$$\begin{aligned}
\text{minimize:} \quad & c_{sea} * q_{7_{sea}} + T^{-1} * \sum_{k=1}^{T} [c_{sea} * \sum_{i=8}^{9} q^k_{i_{sea}} + c_{truck} * \sum_{i=0}^{7} w^k_i \\
& + c_{air_{exp}} * \sum_{i=1}^{7} q^k_{i_{air_{exp}}} + c_{air_{slow}} * \sum_{i=2}^{7} q^k_{i_{air_{slow}}} + h * \sum_{i=0}^{7} I^k_i + b * \sum_{i=0}^{7} S^k_i]
\end{aligned} \tag{2.15}$$

Subject to:

$$\text{(10), (11), (12), (13), and (14)} \quad \ldots \text{for } k \in \{1, 2, 3, \ldots, T\} \tag{2.16}$$

Let v^* be the optimal objective value to the true stochastic program given by (1)-(8), and let $\hat{v}_T$ be the optimal objective value to an SAA problem (15)-(16). It is a well-known result that the SAA solution is an under-estimator of the true stochastic program, that is, $\mathbb{E}[\hat{v}_T] \leq v^*$ (Mak et al., 1999). Further, this under-estimator is monotonically better as number of scenarios in solved in each SAA replication (T) increases, that is, $\mathbb{E}[\hat{v}_T] \leq \mathbb{E}[\hat{v}_{T+1}]$ (Mak et al., 1999). Also, it is well known that $\hat{v}_T$ is a consistent estimator of v^* (Kleywegt et al., 2002), that means,

$$\hat{v}_T \to v^* \quad \text{with probability 1 as } T \to \infty.$$

Thus, using a higher number of scenarios is desirable. However, this increases the size of the integer program to be solved. In general, solving a large integer programming model to optimality can be challenging. To address this computational issue, we use a scenario decomposition method, Progressive Hedging Algorithm (PHA) proposed by Rockafellar and Wets (1991) as part of the SAA scheme. PHA is guaranteed to converge to a global optimal solution in case of convex problems with continuous variables. However, it can be used as a heuristic method in the presence of discrete decision variables (Watson and Woodruff, 2011).

In PHA, one can either solve each scenario sub-problem independently (unbundled-PHA) or create b bundles containing n scenarios each and solve each of these bundles independently (bundled-PHA). To implement unbundled-PHA, we need to reformulate our extensive formulation of the SAA problem as a scenario formulation. In the extensive formulation of the SAA problem given by (15) and (16), Non-Anticipativity Constraint (NAC) is implicitly applied. Stating NAC explicitly leads us to the scenario formulation, which is as follows.

$$\begin{aligned} \text{minimize:} \quad T^{-1} * \sum_{k=1}^{T} [c_{sea} * \sum_{i=7}^{9} q_{i_{sea}}^k + c_{truck} * \sum_{i=0}^{7} w_i^k + c_{air_{exp}} * \sum_{i=1}^{7} q_{i_{air_{exp}}}^k \\ + c_{air_{slow}} * \sum_{i=2}^{7} q_{i_{air_{slow}}}^k + h * \sum_{i=0}^{7} I_i^k + b * \sum_{i=0}^{7} S_i^k] \end{aligned} \tag{2.17}$$

Subject to:

1. Weekly demand must be met in all weeks:

$$\begin{aligned} I_c + \xi_{0_\iota}^k + Q_{2_{sea}} + Q_{0_{air_{exp}}} + Q_{0_{air_{slow}}} - I_0^k + S_0^k = D_0 \ldots \forall k \\ I_0^k + \xi_{1_\iota}^k + Q_{3_{sea}} + q_{1_{air_{exp}}}^k + Q_{1_{air_{slow}}} - I_1^k + S_1^k = F_1 + \xi_{1_\delta}^k + S_0^k \ldots \forall k \\ I_{i-1}^k + \xi_{i_\iota}^k + Q_{(i+2)_{sea}} + q_{i_{air_{exp}}}^k + q_{i_{air_{slow}}}^k - I_i^k + S_i^k = F_i + \xi_{i_\delta}^k + S_{i-1}^k \ldots \forall k \ \& \ i \in \{2,\ 3,\ 4\} \\ I_{j-1}^k + \xi_{j_\iota}^k + q_{(j+2)_{sea}}^k + q_{j_{air_{exp}}}^k + q_{j_{air_{slow}}}^k - I_j^k + S_j^k = F_j + \xi_{j_\delta}^k + S_{j-1}^k \ldots \forall k \ \& \ j \in \{5,\ 6,\ 7\} \end{aligned} \tag{2.18}$$

2. '$x_i^k = 1$' if the quantity of parts received for the week (i + 2) is greater than zero at US port:

$$\begin{aligned} Q_{(i+2)_{sea}} - (M+\varepsilon)*x_i^k &\le 1-\varepsilon \ldots \forall\, k \;\&\; i \in \{0,\, 1, \ldots,\, 4\} \\ q^k_{(j+2)_{sea}} - (M+\varepsilon)*x_j^k &\le 1-\varepsilon \ldots \forall\, k \;\&\; j \in \{5,\, 6,\, 7\} \end{aligned} \tag{2.19}$$

3. '$z_i^k = 1$' if the demand for the week i is greater than on hand inventory comprising of warehouse and inland port:

$$\begin{aligned} I_c + \xi^k_{0_\iota} + Q_{0_{airexp}} + Q_{0_{air_{slow}}} + (M+\varepsilon)*z_0^k &\ge (D_0 - 1) + \varepsilon \ldots \forall\, k \\ I_0^k + \xi^k_{1_\iota} + q^k_{1_{airexp}} + Q_{1_{air_{slow}}} + (M+\varepsilon)*z_1^k &\ge (F_1 + \xi^k_{1_\delta} + S_0^k - 1) + \varepsilon \ldots \forall\, k \\ I^k_{i-1} + \xi^k_{i_\iota} + q^k_{i_{airexp}} + q^k_{i_{air_{slow}}} + (M+\varepsilon)*z_i^k &\ge (F_i + \xi^k_{i_\delta} + S^k_{i-1} - 1) + \varepsilon \ldots \forall\, k \;\&\; i \in \{2,\, 3, \ldots,\, 7\} \end{aligned} \tag{2.20}$$

4. '$w_i^k = 1$' if a trucking trip is made for delivery from US port to the plant, i.e. if both '$x_i^k = 1$' and '$z_i^k = 1$':

$$\left.\begin{aligned} w_i^k &\le x_i^k \\ w_i^k &\le z_i^k \\ w_i^k &\ge x_i^k + z_i^k - 1 \end{aligned}\right\} \ldots \forall\, k \;\&\; i \in \{0,\, 1, \ldots,\, 7\} \tag{2.21}$$

5. Non-Anticipativity constraints:

$$q^k_{7_{sea}} - \widehat{q}_{7_{sea}} = 0 \quad \ldots \forall\, k \tag{2.22}$$

6. Non-negative integer variables:

$$\begin{aligned} \widehat{q}_{7_{sea}} &\in \mathbb{Z}_{\ge 0} \\ q^k_{i_{sea}} &\in \mathbb{Z}_{\ge 0} \quad \ldots \forall k \;\&\; i \in \{7,\, 8,\, 9\} \\ q^k_{i_{air_{slow}}} &\in \mathbb{Z}_{\ge 0} \quad \ldots \forall k \;\&\; i \in \{2,\, 3, \ldots,\, 7\} \\ q^k_{i_{airexp}} &\in \mathbb{Z}_{\ge 0} \quad \ldots \forall k \;\&\; i \in \{1,\, 2, \ldots,\, 7\} \\ I_i^k &\in \mathbb{Z}_{\ge 0} \quad \ldots \forall k \;\&\; i \in \{0,\, 1, \ldots,\, 7\} \\ S_i^k &\in \mathbb{Z}_{\ge 0} \quad \ldots \forall k \;\&\; i \in \{0,\, 1, \ldots,\, 7\} \end{aligned} \tag{2.23}$$

Before we state the unbundled PHA, we define additional notation. We let d denote the set of first-stage and second-stage decision variables for scenario k:

$$d := \{q_{7_{sea}},\ q_{8_{sea}},\ q_{9_{sea}},\ q_{1_{air_{exp}}}, \dots,\ q_{7_{air_{exp}}},\ q_{2_{air_{slow}}}, \dots,\ q_{7_{air_{slow}}},\ z_0, \dots,\ z_7,$$
$$x_0, \dots,\ x_7,\ w_0, \dots,\ w_7,\ I_0, \dots,\ I_7,\ S_0, \dots,\ S_7\}$$

and let $\mathscr{X}^k$ denote the set of feasible solutions for a scenario k. Finally, we define function $f(d)$ as follows

$$f(d) = c_{sea} * \sum_{i=7}^{9} q_{i_{sea}} + c_{truck} * \sum_{i=0}^{7} w_i + c_{air_{exp}} * \sum_{i=1}^{7} q_{i_{air_{exp}}}$$
$$+ c_{air_{slow}} * \sum_{i=2}^{7} q_{i_{air_{slow}}} + h * \sum_{i=0}^{7} I_i + b * \sum_{i=0}^{7} S_i.$$

When using PHA to solve a two-stage stochastic program, the number of iterations required for convergence as well as the quality of the final solution obtained is impacted by the selected value of a penalty factor ρ (Gade et al., 2016). The value of parameter ρ can be either fixed by the decision-maker before running the model or can be computed using the first-stage solutions obtained in the initialization step of PHA. If the value of ρ is fixed by the decision-maker then the decision-maker would either provide a low penalty which would result in slow convergence of PHA and high quality solution or provide high penalty which would result in fast convergence of PHA and low quality solution. If the value of ρ is computed using the first-stage solutions obtained in the initialization step, then the optimal value of ρ can be calculated using the following formula provided in Gade et al. (2016):

$$\rho = \frac{c_{sea}}{\max((T^{-1} * \sum_{k=1}^{T} |q_{7_{sea}}^{k^0} - \widehat{q}_{7_{sea}}^{0}|), 1)}. \tag{2.24}$$

In our implementation of the PHA, we compute the value of ρ using the first-stage solutions obtained in the initialization step. The formal description of the unbundled PHA is given in Algorithm 1.

Algorithm 1: Unbundled PHA

Input :

1. T: Scenario sample size
2. $\xi = (\xi^1,\ \xi^2, \ldots,\ \xi^T)$: Scenario sample vector
3. I: Maximum number of iterations for PHA
4. α: Scalar value greater than 1, used to update the penalty parameter ρ in each successive iteration of PHA

Output: Vector of first stage solution (first stage solution corresponding to each scenario sub-problem)

1 **Initialization:**

2.1 Let $\nu \leftarrow 0$ and $\omega_\nu^k \leftarrow 0$, for $k = 1,\ 2,\ \ldots,\ T$. For each $k = 1,\ 2,\ \ldots,\ T$ compute:

$$d_{\nu+1}^k \in \underset{d \in \mathscr{X}^k}{\arg\min}\ f(d)$$

2.2 $\widehat{q}_{7_{sea}}^0 = T^{-1} * \sum_{k=1}^T q_{7_{sea}}^{k^0}$

2.3 Initialize the value of ρ:

$$\rho_1 = \frac{c_{sea}}{max((T^{-1} * \sum_{k=1}^T |q_{7_{sea}}^{k^0} - \widehat{q}_{7_{sea}}^0|), 1)}$$

2 **Iteration Update:** $\nu \leftarrow \nu + 1$

3 **Aggregation:** $\widehat{q}_{7_{sea}}^\nu \leftarrow T^{-1} * \sum_{k=1}^T q_{7_{sea}}^k$

4 **Price Update:** *For each $k = 1,\ 2,\ \ldots,\ T$ compute:*

$$\omega_\nu^k \leftarrow \omega_{\nu-1}^k + \rho_\nu * (q_{7_{sea}}^k - \widehat{q}_{7_{sea}}^\nu)$$

5 **Decomposition:** *For each $k = 1,\ 2,\ \ldots,\ T$ compute:*

- $d_{\nu+1}^k \in \underset{d \in \mathscr{X}^k}{\arg\min}\ f(d) + \omega_\nu^k * q_{7_{sea}} + \frac{\rho_\nu}{2} * \left\| q_{7_{sea}} - \widehat{q}_{7_{sea}}^\nu \right\|^2$

6 **Updating parameter ρ:** $\rho_{\nu+1} \leftarrow \alpha * \rho_\nu$

7 *If all the scenario solutions $q_{7_{sea}}^k$ are equal or if max number of iterations I are completed, stop. Else, go to step 3.*

In the unbundled PHA, sub-problems for each scenario are solved independently but need to be reconciled. Therefore, the convergence rate of the unbundled PHA is slow. To accelerate the PHA convergence, we can decompose the extensive form of the two-stage stochastic problem by bundles of scenarios (Gade et al., 2016). Using the bundled PHA enables us to solve smaller extensive form formulations of the two-stage stochastic problem (Gade et al., 2016). This also accelerates the convergence rate since we enforce the non-anticipativity constraints (NAC) across all the scenarios considered in a given bundle (Gade et al., 2016), i.e., the solutions obtained from bundles of scenarios need to be reconciled.

Before formally stating the bundled PHA, we will first introduce a problem formulation for a bundle of scenarios. Suppose there are a total of T scenarios in a two-stage stochastic program. The set of T scenarios are partitioned into b bundles containing n scenarios each. Let $\mathscr{B}$ denote the set of all bundles and β be any bundle from the partition ($\beta \in \mathscr{B}$). The probability associated with the bundle β is given by $P_\beta = \sum_{k\in\beta} p_k$, where p_k is the probability of realizing scenario k. The extensive formulation of the problem for a bundle of scenarios β is given as follows.

$$\begin{aligned}\text{minimize:}\quad & c_{sea} * q_{7_{sea}} + \sum_{k\in\beta} \frac{p_k}{P_\beta} * [c_{sea} * \sum_{i=8}^{9} q^k_{i_{sea}} + c_{truck} * \sum_{i=0}^{7} w^k_i \\ & + c_{air_{exp}} * \sum_{i=1}^{7} q^k_{i_{airexp}} + c_{air_{slow}} * \sum_{i=2}^{7} q^k_{i_{air_{slow}}} + h * \sum_{i=0}^{7} I^k_i + b * \sum_{i=0}^{7} S^k_i] \end{aligned} \tag{2.25}$$

Subject to:

$$\text{(10), (11), (12), (13), and (14)} \quad \ldots \forall k \in \beta \tag{2.26}$$

For any bundle β of scenarios, we define the set of decision variables for the extensive formulation associated with the bundle β as d' where

$$\begin{aligned} d' := \{ & q_{7_{sea}},\, q^k_{8_{sea}},\, q^k_{9_{sea}},\, q^k_{1_{airexp}}, \ldots,\, q^k_{7_{airexp}},\, q^k_{2_{air_{slow}}}, \ldots,\, q^k_{7_{air_{slow}}},\, z^k_0, \ldots,\, z^k_7, \\ & x^k_0, \ldots,\, x^k_7,\, w^k_0, \ldots,\, w^k_7,\, I^k_0, \ldots,\, I^k_7,\, S^k_0, \ldots,\, S^k_7\ \forall k \in \beta \} \end{aligned}$$

and let $\mathscr{X}^\beta$ denote the set of feasible solutions for the extensive formulation of the problem associated with

β. Finally, using the set d', we define the function $f(d')$, where

$$f(d') = c_{sea} * q_{7_{sea}} + \sum_{k \in \beta} \frac{p_k}{P_\beta} * [c_{sea} * \sum_{i=8}^{9} q_{i_{sea}}^k + c_{truck} * \sum_{i=0}^{7} w_i^k$$

$$+ c_{air_{exp}} * \sum_{i=1}^{7} q_{i_{air_{exp}}}^k + c_{air_{slow}} * \sum_{i=2}^{7} q_{i_{air_{slow}}}^k + h * \sum_{i=0}^{7} I_i^k + b * \sum_{i=0}^{7} S_i^k]$$

The bundled version of PHA is described in Algorithm 2.

Algorithm 2: Bundled PHA

Input :

1. T: Scenario sample size
2. $\xi = (\xi^1, \xi^2, \ldots, \xi^T)$: Scenario sample vector
3. b: Total number of bundles
4. n: Number of scenarios per bundle
5. I: Maximum number of iterations for PHA
6. α: Scalar value greater than 1, used to update the penalty parameter ρ in each successive iteration of PHA

Note: T = b×n, and T, b, n $\in \mathbb{Z}_{>}$

Output: Vector of first stage solution (first stage solution corresponding to each bundled sub-problem)

1 **Initialization:**

2.1 Let $\nu \leftarrow 0$ and $\omega_\nu^\beta \leftarrow 0, \forall \beta \in \mathscr{B}$. For each β compute:

$$d'^{\beta}_{\nu+1} \in \arg\min_{d' \in \mathscr{X}^\beta} f(d')$$

2.2 $\widehat{q}^0_{7_{sea}} = \sum_{\forall \beta} P_\beta * q^{\beta^0}_{7_{sea}}$

2.3 Initialize the value of ρ:

$$\rho_1 = \frac{c_{sea}}{max((\sum_{\forall \beta} P_\beta * |q^{\beta^0}_{7_{sea}} - \widehat{q}^0_{7_{sea}}|), 1)}$$

2 **Iteration Update:** $\nu \leftarrow \nu + 1$

3 **Aggregation:** $\widehat{q}^\nu_{7_{sea}} \leftarrow \sum_{\forall \beta} P_\beta * q^\beta_{7_{sea}}$

4 **Price Update:** *For each β compute:*

$$\omega_\nu^\beta \leftarrow \omega_{\nu-1}^\beta + \rho_\nu * (q^\beta_{7_{sea}} - \widehat{q}^\nu_{7_{sea}})$$

5 **Decomposition:** *For each β compute:*

- $d'^{\beta}_{\nu+1} \in \arg\min_{d' \in \mathscr{X}^\beta} f(d') + \omega_\nu^\beta * q_{7_{sea}} + \frac{\rho_\nu}{2} * \left\| q_{7_{sea}} - \widehat{q}^\nu_{7_{sea}} \right\|^2$

6 **Updating parameter ρ:** $\rho_{\nu+1} \leftarrow \alpha * \rho_\nu$

7 *If the first stage solutions ($q^\beta_{7_{sea}}$) for every bundled sub-problem is equal or if max number of iterations I are completed, stop. Else, go to step 3.*

2.5 Numerical study

To conduct our numerical study, we analyzed the industrial data we obtained from BMW Manufacturing Company located in Spartanburg, SC. Our industrial collaborators provided us with historical three-year data for actual demand, forecasted demand, and inventory errors for all the parts they procure from their overseas suppliers. The values we use for per unit shipping cost of various transportation modes are based on assumptions due to the sensitivity of the real cost information. The industrial collaborators approved the assumed cost values used for the experimental purposes. However, we do not provide the experimental values of these cost parameters due to confidentiality agreements with the industrial partner.

We studied the forecast errors for one-week to the seven-week ahead and the inventory errors for all parts procured from the overseas region. Upon visualizing the demand data, we observed that the parts could be divided into two categories, namely, low consumption and high consumption parts. A low consumption level means that the given part is seldom required in daily production. A high consumption level means that several boxes of the part are consumed daily to support the production process. The forecast for the low consumption parts was noticeably more inaccurate compared to the high consumption parts. The forecast accuracy for the i week ahead forecast was observed to be more accurate than the $i+1$ week out forecast. Further, the inventory accuracy for the high consumption parts was higher than the low consumption parts.

We selected three parts with high, medium, and low mean forecast errors and two parts, with high and low mean inventory errors. We fitted a probability distribution to all of error terms and observed that the most follow the Johnson's Unbounded distribution with different parameters. Thus, these distributions were used to generate the data used in our study. To construct an SAA problem, sampling can be done in various ways, such as the simple random sampling and stratified sampling using Latin Hypercube Sampling (LHS) (McKay et al., 2000). Using simple random sampling may result in high variance within the SAA scheme, leading to biased results or slowing convergence (Birge and Louveaux, 2011). To address this issue, LHS can be used to reduce the variance in the SAA scheme (Ahmed et al., 2002; Linderoth et al., 2006) or a high number of samples need to be used if one chooses to use simple random sampling. Performing LHS is computationally time-consuming compared to simple random sampling but LHS produces better performing samples at a given sample size. To weigh this trade-off, we decided to test both the sampling methods in our experimental study to investigate the effects of the sampling technique on the quality of the solution obtained and the time required to generate the samples.

It is challenging to solve a two-stage stochastic integer programming problem with a large number of scenarios in a short duration of time. Scenario decomposition-based methods such as unbundled and bundled PHA can be used to reduce the computational time. In our experiments, we solve SAA replications both with and without decomposition-based methods, i.e., unbundled and bundled PHA. For the bundled PHA, we create four bundles of equal size and solve them using distributed computing on four different nodes. The decision to have four bundles in bundled PHA was made due to the limited availability of computational resources. In the case of unbundled PHA, we solve the problem on a single node using series computing.

To study the effects of sampling and decomposition-based methods on the quality of solutions and run time to obtain a solution, we selected a part with medium forecast and high inventory errors. According to our industrial partners (BMW Manufacturing Company) and the data we analyzed, the majority of the parts have medium forecast errors. Further, the industrial partners were more interested in studying the parts with high inventory errors since these parts caused the majority of the operational problems. Therefore, this set of part characteristics is of interest. We run the experiments for the LHS method using all possible combinations of the parameters given in Table 2.3.

Table 2.3: Input parameters values for the experiments with LHS method

Input parameter type	*Value*
Number of scenarios is each SAA replication:	100, 200
Number of SAA replications:	30, 60
Number of evaluation scenarios:	500
Number of evaluation replications:	60

The results obtained using the LHS method are given in Tables 2.4 to 2.9. In these tables, values in the '*Extensive form*' row are obtained by solving the extensive form of the two-stage stochastic program without using any decomposition-based algorithms, whereas the values in '*U-PHA*' and '*B-PHA*' rows corresponds to the values obtained by solving the extensive form of the two-stage stochastic program using unbundled PHA and bundled PHA, respectively.

Table 2.4 provides the estimates of the expected objective value. We observe that the estimate of the objective value obtained using unbundled and bundled PHA is very close to the value obtained by solving the problem without using any decomposition methods. Table 2.5 presents the standard deviation (σ) of the estimate of the objective function value. These results show that with an increase in the number of scenarios in each replication of SAA, the standard deviation value for the estimate of the objective value decreases. The standard deviation values obtained for a given experiment are nearly identical for all solution methodologies. Table 2.6 shows the average time required to solve one SAA replication and the average time required to generate samples per replication of SAA. We see that the average time required to solve one SAA replication and the average time required to generate samples per replication of SAA increases as the number of scenarios increase. Further, the time required to solve the problem without a decomposition method is the lowest and the time required to solve the problem using unbundled PHA is the highest. Table 2.7 provides the values of the optimality gap obtained by solving the problem using different methods. We observe that the optimality gap values obtained by solving the problem using different methods are nearly identical. Table 2.8 and Table 2.9 provide the average number of iterations required for the PHA to converge and the percentage of replications where PHA achieves convergence, respectively. These results indicate that the bundled PHA converges faster and also has higher convergence percentage compared to the unbundled PHA.

Table 2.4: $\mathbb{E}[\hat{v}_N]$ estimate using LHS

Scenarios	100		200	
Replications	30	60	30	60
Extensive form	201401	201644	202837	203062
U-PHA	201410	201654	202844	203069
B-PHA	201412	201651	202839	203063

Table 2.5: σ of $\mathbb{E}[\hat{v}_N]$ estimate using LHS

Scenarios	100		200	
Replications	30	60	30	60
Extensive form	15493	15275	9480	9345
U-PHA	15494	15276	9479	9344
B-PHA	15491	15274	9479	9345

Table 2.6: Average solution time and average sample generation time per replication using LHS (in seconds)

Scenarios	100		200	
Replications	30	60	30	60
Extensive form	0.36	0.37	0.77	0.75
U-PHA	79.15	80.16	203.36	193.56
B-PHA	2.49	2.46	2.92	13.84
Sample generation time	73.61		322.03	

Table 2.7: Optimality gap using LHS

Scenarios	100		200	
Replications	30	60	30	60
Extensive form	7.76%	6.51%	6.05%	5.10%
U-PHA	7.76%	6.50%	6.05%	5.10%
B-PHA	7.80%	6.55%	6.05%	5.10%

Table 2.8: Average number of iterations per replication for PHA using LHS

Scenarios	100		200	
Replications	30	60	30	60
U-PHA	115.47	115.83	148.83	141.63
B-PHA	11.67	11.52	7.47	34.80

Table 2.9: Percentage convergence for PHA using LHS

Scenarios	100		200	
Replications	30	60	30	60
U-PHA	83.33	86.67	80.00	81.67
B-PHA	100.00	100.00	100.00	95.00

From the results discussed so far, we conclude that: (i) the time required to generate samples using the LHS method is considerably high, (ii) solving the problem using bundled PHA and distributed computing is better than solving the problem using unbundled PHA and series computing, and (iii) it is preferable to solve the model without using any decomposition method because of the lower computational time requirement. Based on these observations, we decided to run additional experiments using simple random sampling to reduce the time required to generate the samples in each SAA replication. In this new set of experiments, we solved each SAA replication using the bundled PHA and without using a decomposition algorithm to compare the performance of these two methods as we increased the scenario sample size. We chose bundled PHA since it performed better than unbundled PHA in our experiments. We again chose a part with medium forecast errors and high inventory errors and ran experiments using all possible combinations of the parameters given in Table 2.10.

Table 2.10: Input parameters values for the experiments with simple random sampling

Input parameter type	*Value*
Number of scenarios is each SAA replication:	200, 2000, 3000
Number of SAA replications:	30, 60
Number of evaluation scenarios:	10000
Number of evaluation replications:	60

The results for the experiments run with simple random sampling are presented in Tables 2.11–2.14. In these tables, values in '*Extensive form*' row corresponds to the values obtained by solving the extensive form of the two-stage stochastic program without using any decomposition-based algorithm. For the extensive form solutions, we set the maximum time limit to obtain the solution at five hundred seconds. The values in '*B-PHA*' rows correspond to the values obtained by solving the extensive form of the two-stage stochastic program using bundled PHA.

Table 2.11: $\mathbb{E}[\hat{v}_N]$ estimate using simple random sampling

Scenarios	200		2000		3000	
Replications	30	60	30	60	30	60
Extensive form	200575	201294	206218	207055	207991	208604
B-PHA	200576	201295	206042	207048	207970	208564

Table 2.12: σ of $\mathbb{E}[\hat{v}_N]$ estimate using simple random sampling

Scenarios	200		2000		3000	
Replications	30	60	30	60	30	60
Extensive form	22384	23361	6539	7961	5752	5734
B-PHA	22384	23361	6520	7956	5750	5747

Table 2.13: Average solution time and average sample generation time per replication using simple random sampling

Scenarios	200		2000		3000	
Replications	30	60	30	60	30	60
Extensive form	0.77	0.75	473.00	474.01	506.32	498.67
B-PHA	4.51	5.54	148.04	141.84	307.67	327.15
Sample generation time	0.011		0.089		0.130	

Table 2.11 shows the estimated expected objective function values. We observe that the estimate of the expected objective values obtained using the two solution methods are very close. However, the

Table 2.14: Optimality gap using simple random sampling

Scenarios	200		2000		3000	
Replications	30	60	30	60	30	60
Extensive form	7.77%	6.26%	2.30%	1.57%	1.32%	0.56%
B-PHA	7.77%	6.26%	2.38%	1.58%	1.33%	0.58%

bundled PHA method obtained lower values for the estimate of the expected objective value when the number of scenarios in each replication was greater than two hundred. Table 2.12 presents the standard deviation value (σ) for the estimate of the objective value. The results show that the standard deviation value for the experiments with two hundred scenarios is higher when simple random sampling is used than when LHS is used. Also, the standard deviation values obtained in the case of simple random sampling decrease as we increase the number of scenarios in each SAA replication. Table 2.13 provides the average solution time and average sample generation time per replication using simple random sampling. The time required to generate the scenarios using simple random sampling is significantly lower than using LHS. Additionally, for the experiments with two thousand and three thousand scenarios per replication, the average time required to solve one replication of SAA using bundled PHA is lower than the time required to solve the problem without a decomposition-based method. In Table 2.14, we present optimality gap values. In the experiments with two thousand and three thousand scenarios per replication, the optimality gap values obtained without using a decomposition method are negligibly lower than the optimality gap obtained by bundled PHA. However, solving the problem without using a decomposition method underestimates the optimality gap. When solving the problem without using a decomposition method, we have set a time limit on the solver to obtain a solution. Therefore, in this case, we overestimate the expected objective value, resulting in the underestimation of the optimality gap.

The final set of experiments were run to study the effect of forecast and inventory error uncertainties on the various performance metric, such as the estimate of the objective value and its standard deviation, solution time, and the estimated optimality gap. Generating scenarios using simple random sampling is time-efficient. However, solving the problem with a small number of scenarios generated using simple random sampling results in higher standard deviation values for the estimate of the objective value. Therefore, one needs to solve the problem using a high number of scenarios to reduce the standard deviation for the estimate of the objective value. Without a decomposition method that allows parallel computing, increasing the number of scenarios increases solution time for the two-stage stochastic programming problem. Thus, we run our third set of experiments by generating scenarios using simple random sampling and solving the problem using bundled PHA with distributed computing. All possible combinations of the parameters provided in Table 2.15 were used.

Table 2.15: Input parameters values for the experiments ran to study the effects of different levels of forecast and inventory error uncertainties

Input parameter type	*Value*
Forecast error uncertainty level:	low, medium, high
Inventory error uncertainty level:	low, high
Number of scenarios is each SAA replication:	2000, 3000
Number of SAA replications:	30, 60
Number of evaluation scenarios:	10000
Number of evaluation replications:	60

The forecast and inventory error uncertainty levels are defined using the mean error values. The mean and standard deviation values according to the forecast and inventory error uncertainty levels are described in Table 2.16 and Table 2.17.

Table 2.16: Forecast error uncertainty levels based on mean and standard deviation

Forecast error level	*Mean*	*Standard deviation*
Low	Low	High
Medium	Medium	Low
High	High	Medium

Table 2.17: Inventory error uncertainty levels based on mean and standard deviation

Inventory error level	*Mean*	*Standard deviation*
Low	Low	Low
High	High	High

In Table 2.18, we report the estimate of the $\mathbb{E}[\hat{v}_N]$ and the standard deviation of the estimate obtained for different experiments run for different part types. The results reported in Table 2.18 show that for the same forecast error uncertainty level, the setting with a lower standard deviation of the inventory errors results in a lower estimate of the objective function value and a lower standard deviation for this estimate. Moreover, for the same inventory error uncertainty level, the setting with a lower standard deviation of the forecast error has a lower estimate of the objective value and lower standard deviation for this estimate. In Table 2.19, we report the optimality gap (based on 95% confidence interval) and run time for each setting considered. The minimum optimality gap we obtained is 0.53%, and the maximum is 4.49%. For the same set of experiments, the minimum solution time per replication was 28.79 seconds, and the maximum solution time per replication was 327.15 seconds. Additionally, we report the performance of the B-PHA using the values presented in Table 2.20. The results reported in Table 2.20 show that for the setting considered, the average number of iterations per replication required for bundled PHA to converge was between 4.83 and 10.32. Furthermore, the percentage of replications where bundled PHA achieved convergence varies between 75 % and 100% across all settings. In most settings, convergence was achieved in more than 83 % of replications.

Table 2.18: $\mathbb{E}[\hat{v}_N]$ estimate and σ of $\mathbb{E}[\hat{v}_N]$ estimate

Scenarios		2000				3000			
Replications		30		60		30		60	
Forecast error level	Inventory error level	Estimate	σ	Estimate	σ	Estimate	σ	Estimate	σ
low	low	121358	5407	122200	6451	121213	5075	121954	5242
low	high	292914	9567	296346	10317	296846	8382	296823	7991
medium	low	45043	1251	45051	1351	44879	958	44973	1014
medium	high	206042	6520	207048	7956	207970	5750	208564	5747
high	low	89750	3037	90003	3008	89452	2423	89902	2408
high	high	269905	7218	272462	8298	272364	6351	272890	6343

Table 2.19: Optimality gap and average run time per replication

Scenarios		2000				3000			
Replications		30		60		30		60	
Forecast error level	Inventory error level	Optim -ality gap	Run time (sec)	Optim -ality gap	Run time (sec)	Optim -ality gap	Run time (sec)	Optim -ality gap	Run time (sec)
low	low	2.62%	146.55	1.43%	142.34	2.58%	281.49	1.35%	282.33
low	high	2.56%	146.84	0.99%	148.08	1.05%	283.66	0.59%	271.31
medium	low	4.37%	30.91	4.04%	28.79	4.49%	50.51	3.96%	52.59
medium	high	2.38%	148.04	1.58%	141.84	1.33%	307.67	0.58%	327.15
high	low	3.11%	34.88	2.34%	36.49	3.15%	73.13	2.24%	79.45
high	high	2.12%	132.13	0.87%	141.65	1.11%	287.03	0.53%	280.34

Table 2.20: Average number of iterations per replication and percentage convergence for B-PHA

Scenarios		2000				3000			
Replications		30		60		30		60	
Forecast error level	Inventory error level	Iterat -ions	Conver -gence (%)	Iterat -ions	Conver -gence (%)	Iterat -ions	Conver -gence (%)	Iterat -ions	Conver -gence (%)
low	low	6.83	100.00	6.32	100.00	5.87	100.00	6.02	100.00
low	high	7.97	86.67	8.07	88.33	7.80	90.00	6.43	95.00
medium	low	5.30	100.00	4.83	100.00	5.27	100.00	5.53	100.00
medium	high	8.30	83.33	6.20	95.00	8.27	83.33	10.32	75.00
high	low	5.47	100.00	5.87	100.00	6.30	100.00	6.22	100.00
high	high	7.13	90.00	8.43	85.00	9.10	80.00	8.25	85.00

For obtaining the managerial insights related to the problem we solve, we analyze the optimality gap and run-time per replication values presented in Table 2.19. In general, we can observe that as we increase the number of scenarios or replications, the optimality gap values decrease. However, increases in the number of scenarios in each replication results in an increase in solution time per replication, and an increase in the number of replication results in an increase in the overall solution time required to solve the problem. Thus to find a tradeoff such that a good quality solution is obtained in a reasonable amount of time, the decision-maker can run the experiments with 2000 scenarios and 60 replications for the following reasons, (1.) the solutions obtained using 3000 scenarios and 60 replications are not significantly better than the solutions obtained using 2000 scenarios and 60 replications, (2.) the overall solution time required to solve a problem with 2000 scenarios and 60 replications is approximately equal to the overall solution time required to solve a problem with 3000 scenarios and 30 replications. Further, the overall solution time required to solve a problem with 2000 scenarios and 60 replications is approximately 50% less than the overall solution time required to solve a problem with 3000 scenarios and 60 replications. This is because, on average, solving a single replication of a problem with 3000 scenarios takes twice the time of solving a replication of a problem with 2000 scenarios. Therefore, we suggest that the decision-maker run the experiments with 2000 scenarios and 60 replications.

Next, we compute the Value of Stochastic Solution (VSS) to obtain insights into the cost benefits of solving a stochastic programming model. To compute the VSS solution, we run the two-stage stochastic programming model using SAA with 2000 scenarios and 60 replications because of the argument we discussed in the previous paragraph. We first solve a Mean Value Problem (MVP), in which we solve a stochastic programming model by replacing the values of uncertain input parameters with their respective mean values, and as a result, we obtain an Expected Value Solution (EVS). We next solve a two-stage stochastic programming model with 2000 scenarios and 60 replications and obtain 60 potential candidate solutions. We select a solution with the minimum objective value of 60 solutions we found and call it a Stochastic Value Solution (SVS). Using the EVS and SVS, we solve a problem with 10000 scenarios and 60 replications and record the cost difference observed in each replication resulting from using the EVS solution against the SVS solution (i.e., Value of EVS - Value of SVS). The VSS results obtained by running the experiments for different part types with 10000 scenarios and 60 replications are presented in Table 2.21. The VSS savings reported in Table 2.21 are the expected savings observed per week on a recurrent basis.

Table 2.21: Value of stochastic solution

Forecast error level	Inventory error level	Estimate of upper bound (v^*) using SVS	Objective value estimate using EVS	Estimate for VSS	Percent savings using VSS
low	low	121640.88	132616.16	10975.28	8.28%
low	high	295671.73	322791.88	27120.15	8.40%
medium	low	46439.66	76415.81	29976.15	39.23%
medium	high	207448.71	231735.94	24287.22	10.48%
high	low	91064.31	119405.35	28341.04	23.74%
high	high	271820.27	309186.27	37365.99	12.09%

From the values in Table 2.21, we observe that a minimum of \$10,975.28 and a maximum of \$37,365.99 possible expected savings can be obtained every week from solving a stochastic model. Correspondingly, we observe that a minimum of 8.28% and a maximum of 39.23% possible expected savings can be obtained by solving a stochastic model. The results show that the lowest savings are obtained in the case of low forecast and inventory error levels. This is because when the error levels are low, it is reasonable to solve the mean value problem instead of the stochastic problem. Further, as discussed earlier, most parts that the manufacturer procures fall under the category of medium forecast and high inventory error level. For the parts falling in this category, the savings can be significantly improved by improving the inventory records (i.e., reducing inventory error level by increasing the inventory accuracy).

2.6 Conclusions

In this chapter, we formulated a two-stage stochastic integer programming model to reduce the long-term expected material procurement cost for a manufacturing company while considering forecast and inventory error uncertainty. By reviewing the existing literature related to supply chain planning under uncertainty, we observe that the current studies have not considered uncertainty in inventory level in their problem settings. Inventory uncertainty exists in the majority of the real-world manufacturing problem setting with inventories.

We studied the effects of scenario sampling methods and solution methodologies on the quality of the solution and computational time for the two-stage stochastic integer programming method. For the problem under consideration, we showed that generating a large number of scenarios using simple random sampling and solving the problem using bundled PHA is preferable. Using bundled PHA helps us decompose the large-scale problem into easy-to-solve smaller sub-problems. Further, these smaller sub-problems are solved independently at each iteration using distributed computing to reduce the computational time. In the final set of experiments, we also studied the effects of varying the forecast and inventory error uncertainty levels on solution quality, solution time, and the bundled PHA performance. From the results obtained by running the final set of experiments, we observe that the estimate of the expected objective value and the standard deviation of the estimate increases with an increase in the standard deviation of the uncertain input parameter.

2.7 Potential future research

In this chapter, we solved the extensive form of the two-stage stochastic programming problem using the PHA. In the algorithm we used, the lower bound information is not computed in any step. Even without lower bound information, the PHA generates a high-quality solution (Gade et al., 2016). Having the lower bound information in PHA can help us form the termination criteria and check the quality of the solution with respect to the actual optimal objective value. We tried to compute the lower bounds at every iteration by using the method proposed by Gade et al. (2016). However, the method proposed by Gade et al. (2016) did not yield tight lower bounds for our problem. Future research work can be done to improve this aspect. A potential idea to address this problem is discussed below.

In every *Decomposition* step of the PHA, we are solving the augmented lagrangian problem. Additionally, the first-order Lagrange multiplier (ω) is updated in *Price Update* step and the second-order Lagrange multiplier (ρ) in every *Updating parameter* ρ step. We can modify this problem such that in every *Decomposition* step, we are solving a first-order Lagrangian relaxation problem. This modification can be achieved by changing the non-anticipativity constraint (NAC) such that the absolute deviation of the first-stage scenario solutions with respect to the aggregate of the first-stage scenario solution is equal to zero. We refer to this revised NAC by 'modified NAC'. We can have a linear integer programming problem by linearizing the modified NAC, which can be done trivially. A first-order Lagrangian problem can be obtained by dualizing the linearized version of modified NAC. Using this information, we can replace the second-order objective function with the first-order objective function in the *Decomposition* step of the PHA. Another modification can be made to the dual price update step in PHA. The Lagrangian multipliers (dual prices) for the first-order Lagrangian problem can be obtained by using the sub-gradient method as discussed in the Fisher (2004). We can use a similar way to update the dual prices in every PHA iteration. We believe these modifications can help us compute the tight lower bounds for the PHA.

Chapter 3

Optimal Sea-container Consolidation under Uncertainty and Vendor Hub Inventory Control

3.1 Introduction

For a manufacturing company that acquires parts and materials from overseas, there can be substantial time between the placement and receipt of replenishment orders due to long lead times. Thus, companies often need to rely on forecasted line-side demand when placing orders. The differences between the actual demand for parts and the quantities forecasted well in advance might lead to part shortages or overflows in the warehouse, which are both costly to the manufacturer. As a result, it is of interest to decrease such discrepancies and minimize warehouse utilization while satisfying the demand. Supply chain infrastructure such as consolidation centers or repackaging facilities can be used to better match the mix of products in each shipment of parts to the expected demand.

We consider a manufacturer with an overseas consolidation center that can combine shipments from multiple inbound trailers to load sea-containers with a *favorable* parts mix that helps reduce the manufacturer's warehouse utilization. These sea-containers are then delivered to the manufacturer using a combination of maritime, rail, and road transportation. A sea-container contains multiple parts after the consolidation process. Further, a sea-container is always unloaded entirely by the manufacturer at the destination warehouse. Therefore, when a sea-container is unloaded to restock a part whose inventory levels are low, other parts that are in the same container but not needed to be restocked are also unloaded, filling the warehouse with unnecessary parts. However, the warehouse space utilization needs to be maintained below a defined level to ensure a smooth production process. Therefore, it is desirable to have a load-mix (i.e., a mixture of various parts) in the sea-containers to avoid increasing warehouse utilization unnecessarily.

We assume that the orders for parts are placed using the weekly forecasted demand, whereas the manufacturers' warehouse is designed to carry less than a week's part supply. Therefore, we compare the effect of using the weekly and daily forecasted demand for consolidation on the manufacturing warehouse utilization. We assume that the manufacturer considered in this chapter does not have a daily demand forecasting model. Consequently, in this chapter, we build and study the time-series forecasting models to predict the daily demand quantities.

We observe the random errors in predicted and actual daily demand values from the time-series study, resulting in excess inventory or shortages at the manufacturing warehouse. We assume that the suppliers considered in this study must maintain an agreed inventory level of the part to fulfill any emergency orders arising due to shortages at the manufacturer. Thus, we build and study the chance-constraint optimization model to determine the optimal supplier inventory levels for emergency orders under uncertain forecast errors.

The rest of this chapter is organized as follows, in *Section 3.2* we review the existing literature on shipment consolidation, demand forecasting and inventory management, and inventory replenishment. In *Section 3.3* we define the problem and describe the solution approach. A discussion on time-series models built and the results obtained is presented in *Section 3.4*. The chance-constraint optimization problem formulation is discussed in *Section 3.5* and the solution methodology along with experimental results are provided in *Section 3.6*. Concluding remarks are provided in *Section 3.7*, and the potential future research is discussed in *Section 3.8*.

3.2 Literature review

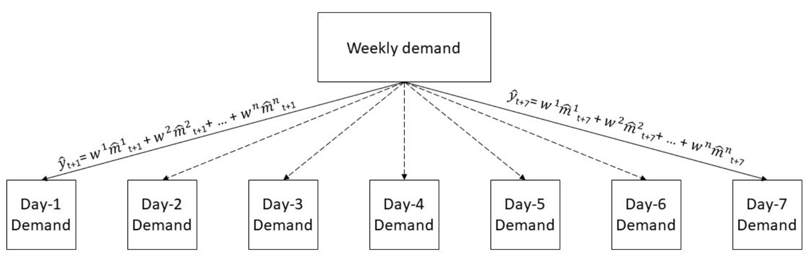
We first review the literature that studies consolidation problems. Tyan et al. (2003) studied a shipment consolidation problem for a global supply chain using air-freight as the transportation mode. They aim to minimize the total cost due to linehaul, consolidation operations, inventory holding, penalty for late shipments, and lost capacity by determining the optimal quantity for shipment mode picked on each day to be shipped by each flight. A freight consolidation problem considering multiple customers, multiple intermediate (consolidation) facilities, and a single consolidation facility was studied by Hanbazazah et al. (2019) and Hanbazazah et al. (2020). Hanbazazah et al. (2019) studies a freight consolidation problem with divisible shipments to minimize the total cost of shipping by solving a Mixed Integer Linear Program (MILP). Hanbazazah et al. (2020) considers a freight consolidation problem with indivisible shipments to minimize the total cost. The divisible shipments considered in Hanbazazah et al. (2019) can be divided into multiple subloads and transported from one consolidation facility to another, whereas the indivisible shipments considered in Hanbazazah et al. (2020) cannot be. Zhang et al. (2021) studied a consolidation problem to optimally consolidate the sub-orders originating from multiple warehouses via a transshipment among warehouses. The authors formulated a multi-commodity network flow problem to reduce the order fulfillment cost by determining the consolidation warehouse for each order and transshipment for each suborder. All these studies considered the order demand as a deterministic input to the optimization model along with other input parameters, and solved a cost minimization problem. Sonntag et al. (2021) studied an inventory routing problem with time-based shipment consolidation to minimize the cost under stochastic demand. The authors formulate a chance constraint program to determine an optimal set of retailer groups, the shipment interarrival times for each group, and the base stock quantity for each supplier. The authors consider minimizing inventory holding costs along with other costs in the objective function. However, minimizing the holding cost is not the objective of the end customer (manufacturer) in our case. In our problem, the manufacturer is more interested in minimizing the daily warehouse utilization to support a smooth production process. Therefore, the quantity of the particular material packed in a container is important for the manufacturer. As a result, we formulate a problem to determine the daily demands of the manufacturer corresponding to the week the container was built to satisfy the demand. Using these forecasted daily demands, the manufacturer and its third-party logistics (3PL) service providers can solve transshipment consolidation problems to minimize the cost and pack the containers using the forecasted daily demands instead of forecasted weekly demands. Similar to the paper by Hanbazazah et al. (2019), we assume all the orders considered for this study to be divisible. However, in our problem, there is only one consolidation center available.

To predict the daily demand quantities, we build a variety of time-series forecasting models. A comprehensive comparison study of univariate time-series models using Prophet and ARIMA and multi-variate time series using Lasso and LSTM (Long Short Term Memory) was done by Motamedi et al. (2021) to forecast the demand for platelet usage. In a univariate method, only demand is used as an input parameter to train the time-series models. In a multi-variate method developed to forecast demand for platelets, clinical indicators are also used as input parameters in addition to demand. In this study, the authors demonstrate that when a sufficient amount of data is available, the traditional methods like ARIMA perform as good as the machine learning-based multi-variate methods. In this study, we only use univariate methods because we assume that the manufacturer only has historical demand data. Acar and Gardner Jr (2012) performed a time-series forecasting study to select a forecasting method based on the operational performance (i.e., the total cost incurred and customer service level) in the supply chain. The authors studied three different exponential smoothing methods: simple exponential smoothing, Holt's additive model, and the damped additive trend model. The experimental results suggest that the damped trend method performs best with respect to the operational performance metrics. In this study, we build our forecasting models by considering Simple Moving Average, Simple Exponential Smoothing, SARIMA, and Prophet methods. Li et al. (2021) and Guo et al. (2021) studied the ensemble-based forecasting models. Li et al. (2021) did a comparison study using various time-series models and suggested that the ensemble model that uses STL (Seasonal and Trend decomposition using Loess) and XGBoost method is simpler to apply while generating good quality forecasts. The STL method is used to predict the seasonal and trend component, while the XGBoost is used to predict the residuals. The authors use the STL method because they observe an irregular seasonality over time. In contrast, the analysis of the data provided by our industry partner shows that the seasonality is regular over time. Thus, we chose not to test an ensemble model built using STL and XGBoost for our problem. Guo et al. (2021) proposed an ensemble-based forecasting model using Prophet and SVR (Support Vector Regression) to predict demands in the manufacturing industry with seasonality. The Prophet method predicts the linear structures, while the residuals are predicted using the SVR. The final forecast is generated by summing the linear structures and residual components. Similar to Guo et al. (2021), we use the grid search method to select the best values for the hyperparameters in the Prophet method. However, unlike Li et al. (2021) and Guo et al. (2021), we build our ensemble models by using simple average and weighted average methods. A detailed discussion on the ensemble model used in this study is provided in Section 3.4.

Multiple approaches such as the *traditional* approach, *Vendor Managed Inventory (VMI)* approach, and *Vendor Hub* approach are used to control the inventory levels across different areas within a supply chain (Lee and Chu, 2005). In the *traditional* approach, the downstream partners in a supply chain decide on the stock and hold the inventory. In the *VMI*, the supplier determines the stock levels and replenishment cycles of the downstream partners. In the *Vendor Hub* approach, the supplier maintains a minimum agreed stock level of a product at its warehouse, and the manufacturer only pays for the amount of stock consumed. Burke et al. (2009), Bilsel and Ravindran (2011), and Moheb-Alizadeh and Handfield (2018) studied a sourcing problem under uncertainty for the *traditional* approach. In Burke et al. (2009), the authors studied the problem of maximizing the expected profits under uncertain demand and supply by determining the optimal set of suppliers selected for order placement, the total order quantity, and allocations of order quantities to each selected supplier. A multi-objective chance constraint optimization model was formulated by Bilsel and Ravindran (2011) to minimize the cost, maximize the quality, and minimize the lead times by optimally determining the primary and backup suppliers for each product and the amount of each product to be shipped using each supplier. The authors considered uncertainties related to the capacity of suppliers, customer demands, and transportation costs in the problem formulation. Further, all uncertain quantities were assumed to follow a normal distribution, and the multi-objective optimization model was solved using the goal programming method. Moheb-Alizadeh and Handfield (2018) also studied a multi-objective chance-constraint optimization model for supplier selection and order allocation. However, the authors only considered the demand to be stochastic. The objectives of the problem was to minimize the cost, minimize carbon emissions, and maximize the firm's social responsibility by optimally selecting the suppliers and allocating orders to them. Govindan (2015) compared the performance of the traditional and the VMI approaches under stochastic demand using the total supply chain cost as a performance measure. Via a numerical study, the authors observed that using the VMI approach results in lower costs compared to the traditional approach. The manufacturer we consider uses the *vendor hub* approach for inventory management. In particular, the manufacturer places the orders with its suppliers using the forecasted demand quantities and pulls the material. Further, the supplier must maintain the agreed inventory levels at all times. The manufacturer utilizes this excess inventory stored with the supplier to fulfill any emergency orders that may arise due to a stock-out. While there are a plethora of studies done assuming a *traditional* or *Vendor Managed Inventory (VMI)* approach, to the best of our knowledge, no study has been done considering the *vendor hub* approach. In this study, we assume that only one supplier is available to fulfill the order for a particular material, and we determine the order quantities (according to the final consolidation quantities) using forecasted demands. A problem of inventory replenishment considering emergency orders was studied by Poormoaied et al. (2020) and Poormoaied et al. (2022). Poormoaied et al. (2020) studied a problem of time-based inventory replenishment and emergency orders, and Poormoaied et al. (2022) studied a quantity-based inventory replenishment and emergency orders. In both the studies, the customer demand is considered to arrive as per the Poisson process, and the supplier is assumed to have infinite capacity. In practice, obtaining an infinite capacity supplier is not possible because the supplier also needs to maintain or lower its operating costs. Therefore, we assume that time when the manufacturer will place the orders for the emergency shipments is known and determine the excess inventory level the supplier must maintain to fulfill these emergency orders.

3.3 Problem description and solution approach

We collaborated with BMW Manufacturing Company, located in Spartanburg, South Carolina, to conduct the numerical study. The manufacturer has the deterministic daily demand known for the next fourteen days, and after that, the weekly forecasted demand is available every week for up to one year from the current day. The manufacturer provided us with the historical actual daily demand values for twelve different parts for the years 2018 and 2019. Further, the manufacturer also provided us with the two years (i.e., 2018 and 2019) of historical forecasted weekly demand values up to seven weeks in the future from the forecast date.



Notation	Description
$\hat{m}^1_{t+i}$	Forecast generated for *Day-i* after *t* time periods using individual *model-i*
w^1	Weight assigned to forecasting *model-i*
$\hat{y}_{t+i}$	Final forecast for *Day-i* after *t* time periods
n	Total number of individual models used

Figure 3.1: Splitting weekly order quantities into daily

The manufacturer uses sea-freight as the primary mode of transportation for procuring materials from the overseas region. We assume that the lead-time using the sea-freight is seven weeks. This lead-time can be further broken down as follows: two weeks for preparing and delivering the order from supplier to consolidation center in Europe, one week for delivering the material from consolidation center to the port in Europe, twelve days on the ship from the European port to the US Port, four days on the rail from the US port to the manufacturing plant and twelve days of safety lead-time. Therefore, when a new sea-freight order is placed, it is expected to satisfy the demand in the eighth week.

Due to the considerably long lead-time discussed above, the sea-containers are packed at the consolidation facility five weeks before their consumption week, when no daily demand information is available. However, the manufacturer keeps track of the historical daily demand for all parts. Using this data, we build the time-series forecasting model that helps the manufacturer predict the daily demands in the week for which the material is being packed into the sea-container at the overseas consolidation facility. In general, the demand forecast generated for the *week* i is more accurate than the forecast generated for the *week* $i+j$. Therefore, we assume that the order quantities placed by the manufacturer can be adjusted at the time of consolidation because the additional required material can be procured on an emergency basis, or the excess material can be held by the cross-dock facility or returned to the supplier. Accordingly, the forecasted demand quantities obtained during the consolidation are the final quantities using which the sea-containers are packed.

The predicted daily demand values obtained using time-series models are subject to forecasting errors. As a result of these errors, there will be either excess inventory in the manufacturer's warehouse at the end of the production day, or part shortages will impact manufacturing operations. The excess inventory in the warehouse at the end of a production day is undesirable. However, shortages are detrimental because they can lead to complete production line stoppage, which is very expensive. Suppose a material shortage is anticipated, and it is infeasible to fulfill the emergency demands arising from shortages using sea-freight due to its high transportation lead-time; in such circumstances, the material is expedited using a costly air-freight option with a short transportation lead-time. To ensure the required material is always available at the overseas supplier in case of such emergencies, the manufacturer has an agreement with the supplier such that the supplier must always maintain the agreed amount of material available in their warehouse for emergencies.

Since the shortages and the excess inventory are undesirable, we study the error distribution of the predicted demand values. Using this error distribution, we formulate a chance constraint optimization model to obtain the order quantity values such that the demand is met completely in (1-ε)% scenarios. Whereas, in the rest ε% scenarios, shortages are allowable. Our objective in this optimization problem is to reduce the amount of excess and shortage inventory at the end of each day across all the realized scenarios for a given week. The decision variable is the order quantity corresponding to each day of the week. The optimal order quantity corresponding to each day obtained by solving the optimization problem can be used by the manufacturer to effectively manage the amount of inventory maintained by the supplier for emergency orders.

3.4 Time series forecasting study

In this section, we discuss the time-series forecasting model we built to predict the daily values in the sixth week (35^{th} to 42^{nd} days, i.e., when the sea-containers would be packed at the consolidation center) from the current day (the time when the order is placed with a supplier). We start by building individual models, such as simple moving average and simple exponential smoothing, which do not account for trend and seasonality components in the time-series data. Then we progress toward building time series forecasting models using triple exponential smoothing (aka Holt-Winters method), SARIMA (Seasonal Autoregressive Integrated Moving Average), and Prophet methods (Facebook Open Source, 2017), which account for trend and seasonality in the time-series data. Subsequently, we build our final time-series model using an ensemble approach by combining different individual models.

In the simple moving average model (SMA) the average of the last x periods from the current period n is taken to predict the value at period $n+1$ for the SMA model with order x. On the contrary, the simple exponential smoothing (SES) method uses a smoothing parameter α, which influences the weights given to the more recent and distant past observations. A large alpha value indicates more weight is given to the recent past observations. Neither SMA or SES account for trend and seasonality components in time-series data. Therefore, the forecast values generated by SMA and SES tend to be less accurate for the time-series data which contains trend and seasonality factors. If the time-series contains the trend and seasonality components, methods such as triple exponential smoothing (TES) and SARIMA can be used. Compared to SES, TES uses two additional smoothing parameters to account for trend and seasonality. ARIMA (Autoregressive Integrated Moving Average) has three parameters autoregression order (p), differencing factor (d), and moving average order (q). ARIMA model is suitable for forecasting when the time-series data has no seasonality component. Seasonal ARIMA (SARIMA) is favorable when the time-series data has a seasonality component. Compared to ARIMA, SARIMA has four additional parameters seasonal autoregression order (P), seasonal differencing factor (D), seasonal moving average order (Q), and seasonality period (m). Hyndman and Athanasopoulos (2018) provides a comprehensive explanation of these forecasting models. In addition to the SMA, SES, TES, and SARIMA methods, we also build a forecasting model using Prophet. The Prophet is an open-source package developed by Facebook for forecasting.

We build the time-series forecasting models in Python programming language using SMA, SES, TES, and SARIMA methods. The pandas package is used to build the SMA models with different orders, the statsmodels package is used to build the SES and TES models, the pmdarima package is used to build the

SARIMA model, and the prophet package is used to build the Prophet model. We use different orders for the SMA model, such as 2, 3, 4, 7, 14, and 28, and select the model with an order with the least root mean squared error on the training data for forecasting. We train the SES model using different α values, such as 0.66, 0.5, 0.4, 0.25, 0.13, and 0.07, along with optimized SES in which the statsmodel package determines an appropriate α value automatically. We select one SES model with the least root mean squared error on the training data for forecasting. Forecasting using the TES model is very trivial using the statsmodel package, we only need to specify the seasonality period and the type of trend and seasonality. For the forecasting model developed using the SARIMA method, appropriate values for parameters p, q, P, and Q were searched using the stepwise algorithm (Hyndman and Khandakar, 2008) in pmdarima package.

Prophet forecasting package is also available in Python in addition to the traditional methods discussed above. We referred to Facebook Open Source (2017) for building the forecasting model using Prophet. To determine the appropriate values for the *changepoint_prior_scale* and *seasonality_prior_scale* hyperparameters of the Prophet model, we use a grid search.

A single forecasting method cannot work satisfactorily for all parts. Further, the appropriate method can change from time to time depending on the changes in the time-series data. Therefore, an ensemble model is often created by combining different forecasting methods. Figure 3.2 provides a visual representation of the ensemble forecasting model. The input time series data is used to train n different individual models (i.e., SMA, SES, TES, etc.). The forecasted values generated by these individual models are then given as an input to the generalizer module, which converts these multiple-input values into a single forecast value according to a predetermined rule. We develop three ensemble forecasting models by blending (1.) TES and SARIMA, (2.) TES, SARIMA, and Prophet, and (3.) SARIMA and Prophet. The *generalizer* generates the final forecast value for each ensemble model by either taking (1.) a simple average of the forecasted values by each individual model or (2.) a weighted average of the forecasted values by each individual model. As a result, we evaluate six ensemble models, three using a simple average and three using a weighted average.

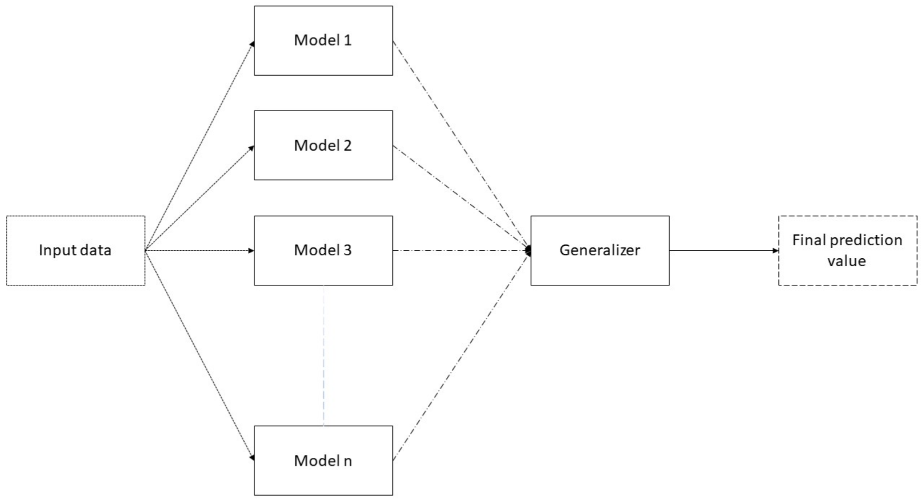


Figure 3.2: Ensemble forecasting model to predict daily demand values

In the case of an ensemble model created using weighted averages, the weight for each individual model was determined according to its error score. Particularly, we first compute an error score on the training data either using the MAPE (Mean Absolute Percent Error) or RMSE (Root-Mean-Square Error) techniques for every individual method (i.e., TES, SARIMA, Prophet). Then, we normalize these errors so that the sum of errors for all the individual models is one. Subsequently, we subtract the normalized error scores from one for each method to get the accuracy score. The final weights are then computed by normalizing the accuracy score.

Our industrial partner provided us with two years of daily demand data for twelve different parts. The twelve parts can be categorized as follows; six were high runners (i.e., the parts with a high amount of daily consumption), and six were low runners (i.e., the parts with a low amount of daily consumption). We label these distinct high runners and low runners parts as $high\ runner - i$ and $low\ runner - i$, for $i \in \{1, 2, \ldots, 6\}$, respectively. We select three of each high runner and low runner parts for our first set of experiments that are done to determine the final experimental setup. Later, for the final study, we use the data for all twelve parts to evaluate the performance of candidate forecasting models. We use the first 625 days of data for training our time series models, and the last 90 days of data are used for testing purposes. We run (train) all the models weekly and generate the daily forecast for the sixth week (35th to 42nd days). We do not consider holiday weeks, such as Thanksgiving and Christmas, in the numerical study. Therefore, we have a total of 13 weeks for which we test our forecasting models.

The initial numerical study is done to determine the error estimator and the training setup we will use to evaluate different forecasting models, and the final study focuses on this evaluation. In the initial study, the error scores for the individual and ensemble forecasting models are computed by generating the forecast values using five different experimental schemes for three different high runner and three different low runner parts. The experimental schemes that we use are as follows: (1.) Use 60 weeks of recent past data to train the forecasting model and compute the errors on the training model using MAPE (60 Rolling - MAPE), (2.) Use 89 weeks of recent past data to train the forecasting model and compute the errors on the training model using MAPE (89 Rolling - MAPE), (3.) Use 89 weeks of recent past data to train the forecasting model and compute the errors on the training model using RMSE (89 Rolling - RMSE), (4.) Use all available data until the current time period to train the forecasting model and compute the errors on the training model using MAPE (MAPE), and (5.) Use all available data until the current time period to train the forecasting model and compute the errors on the training model using RMSE (RMSE).

The abbreviations used in column headings of Tables 3.2 – 3.8 are defined in Table 3.1. The results for the initial forecasting study, are provided in Table 3.2 – 3.5. Table 3.2 and 3.3 present the average daily MAPE values obtained by running the trained individual forecasting models to make predictions on the test data of different high runner and low runner parts. We observe that, in general, the forecasted values obtained using SMA and SES methods have high average daily MAPE scores when compared to TES, SARIMA and Prophet methods. Further, we observe that lower MAPE values can be achieved using the experimental schemes *89 Rolling - MAPE* or *89 Rolling - RMSE*.

Table 3.1: Description of the abbreviations used in Tables 3.2 – 3.8

Notation	Description
SMA	Simple moving average
SES	Simple exponential smoothing
TES	Triple exponential smoothing
SARIMA	Seasonal autoregressive integrated moving average
SA TS	Simple average ensemble using TES and SARIMA
WA TS	Weighted average ensemble using TES and SARIMA
SA TSP	Simple average ensemble using TES, SARIMA and Prophet
WA TSP	Weighted average ensemble using TES, SARIMA and Prophet
SA SP	Simple average ensemble using SARIMA and Prophet
WA SP	Weighted average ensemble using SARIMA and Prophet

Table 3.2: Average of daily MAPE values for individual models obtained using different schemes for high runner parts

Method	Material number	SMA	SES	TES	SARIMA	Prophet
60 Rolling - MAPE	High Runner - 3	129.79	124.81	110.45	104.64	95.46
89 Rolling - MAPE	High Runner - 3	130.49	129.93	105.92	88.98	80.38
89 Rolling - RMSE	High Runner - 3	130.49	129.93	105.76	88.98	81.17
MAPE	High Runner - 3	129.80	126.34	110.51	92.32	88.54
RMSE	High Runner - 3	129.80	126.34	110.28	92.32	88.58
60 Rolling - MAPE	High Runner - 5	90.15	91.39	84.11	69.28	58.46
89 Rolling - MAPE	High Runner - 5	90.15	91.29	92.82	57.80	65.29
89 Rolling - RMSE	High Runner - 5	90.15	91.84	93.40	58.70	65.29
MAPE	High Runner - 5	90.15	91.59	85.53	57.86	68.52
RMSE	High Runner - 5	90.15	91.81	89.00	57.57	68.58
60 Rolling - MAPE	High Runner - 6	77.82	75.18	48.55	52.40	60.79
89 Rolling - MAPE	High Runner - 6	77.82	75.62	49.07	51.25	68.99
89 Rolling - RMSE	High Runner - 6	77.82	75.62	49.06	51.25	69.48
MAPE	High Runner - 6	77.82	74.77	48.59	50.35	70.63
RMSE	High Runner - 6	77.82	74.77	48.62	50.45	70.59

Table 3.3: Average of daily MAPE values for individual models obtained using different schemes for low runner parts

Method	Material number	SMA	SES	TES	SARIMA	Prophet
60 Rolling - MAPE	Low runner - 1	79.89	79.86	56.65	50.93	50.79
89 Rolling - MAPE	Low runner - 1	79.89	78.82	60.02	56.30	57.62
89 Rolling - RMSE	Low runner - 1	79.89	78.98	58.62	56.25	57.65
MAPE	Low runner - 1	79.89	78.67	64.18	49.71	50.17
RMSE	Low runner - 1	79.89	78.67	63.97	49.71	50.20
60 Rolling - MAPE	Low runner - 3	148.67	153.39	135.62	140.27	145.92
89 Rolling - MAPE	Low runner - 3	148.67	156.18	140.47	158.58	133.12
89 Rolling - RMSE	Low runner - 3	148.67	156.18	140.47	158.58	132.15
MAPE	Low runner - 3	148.67	153.67	146.22	159.56	137.32
RMSE	Low runner - 3	148.67	153.67	146.22	159.81	135.89
60 Rolling - MAPE	Low runner - 6	74.94	74.03	47.07	49.18	52.07
89 Rolling - MAPE	Low runner - 6	74.94	73.98	47.22	47.42	57.05
89 Rolling - RMSE	Low runner - 6	74.94	73.98	47.18	47.45	57.01
MAPE	Low runner - 6	74.94	73.98	47.10	47.24	58.19
RMSE	Low runner - 6	74.94	73.98	47.05	47.25	58.23

Tables 3.2 and 3.3 show that not a single individual TES, SARIMA, and Prophet model has lowest average daily MAPE scores for all parts. Therefore, we also evaluate ensemble models that combine these forecasting models. Tables 3.4 and 3.5 present the average daily MAPE values obtained by running the trained ensemble forecasting models to make predictions for different high runner and low runner parts.

Table 3.4: Average of daily MAPE values for ensemble models obtained using different schemes for high runner parts

Method	Material number	SA TS	WA TS	SA TSP	WA TSP	SA SP	WA SP
60 Rolling - MAPE	High Runner - 3	100.06	102.05	95.99	96.92	94.34	94.82
89 Rolling - MAPE	High Runner - 3	95.83	96.11	87.02	87.24	82.33	82.13
89 Rolling - RMSE	High Runner - 3	95.83	93.21	86.71	86.94	82.34	82.34
MAPE	High Runner - 3	99.35	98.74	91.88	92.00	87.39	87.48
RMSE	High Runner - 3	99.35	96.77	91.63	91.61	87.98	86.93
60 Rolling - MAPE	High Runner - 5	74.94	77.03	67.99	68.06	61.43	59.65
89 Rolling - MAPE	High Runner - 5	72.56	74.66	67.33	67.82	60.46	61.09
89 Rolling - RMSE	High Runner - 5	73.28	73.71	67.92	68.20	60.73	60.67
MAPE	High Runner - 5	69.84	71.13	64.62	64.95	61.60	62.42
RMSE	High Runner - 5	71.16	71.52	66.31	66.51	61.48	61.50
60 Rolling - MAPE	High Runner - 6	50.13	49.61	52.86	52.65	56.04	56.87
89 Rolling - MAPE	High Runner - 6	49.97	49.67	54.91	53.41	59.20	57.89
89 Rolling - RMSE	High Runner - 6	49.90	49.83	55.01	54.99	59.44	60.15
MAPE	High Runner - 6	49.37	49.13	54.97	53.37	59.35	57.25
RMSE	High Runner - 6	49.43	49.35	55.00	54.97	59.42	60.25

Table 3.5: Average of daily MAPE values for ensemble models obtained using different schemes for low runner parts

Method	Material number	SA TS	WA TS	SA TSP	WA TSP	SA SP	WA SP
60 Rolling - MAPE	Low runner - 1	51.35	52.51	49.42	49.49	50.06	50.14
89 Rolling - MAPE	Low runner - 1	52.29	52.52	51.68	51.66	54.98	55.04
89 Rolling - RMSE	Low runner - 1	51.65	51.89	51.42	51.41	55.00	55.01
MAPE	Low runner - 1	54.04	54.32	50.65	51.06	48.59	48.55
RMSE	Low runner - 1	53.93	54.28	50.49	50.57	48.57	48.63
60 Rolling - MAPE	Low runner - 3	135.83	136.04	139.02	138.95	141.50	143.33
89 Rolling - MAPE	Low runner - 3	149.57	149.30	142.92	143.78	144.61	144.13
89 Rolling - RMSE	Low runner - 3	149.57	149.48	143.34	144.59	144.75	144.38
MAPE	Low runner - 3	151.26	151.09	146.98	146.98	148.29	147.24
RMSE	Low runner - 3	151.92	151.99	146.55	148.92	147.49	147.36
60 Rolling - MAPE	Low runner - 6	48.03	47.95	48.97	48.92	50.25	50.34
89 Rolling - MAPE	Low runner - 6	47.25	47.23	49.56	48.90	51.18	50.11
89 Rolling - RMSE	Low runner - 6	47.25	47.22	49.56	49.66	51.17	51.59
MAPE	Low runner - 6	47.09	47.09	49.74	49.01	51.52	50.18
RMSE	Low runner - 6	47.08	47.05	49.75	49.83	51.52	51.99

Based on the data obtained from individual and ensemble models, reported in Tables 3.2 – 3.5, we observe that lower MAPE values can be achieved using either the *89 Rolling - MAPE* or *89 Rolling - RMSE* scheme. Additionally, a smaller amount of training data is required to train the forecasting models using *89 Rolling - MAPE*. Using a smaller data set means that potentially lower training time is required to build the forecasting models. Therefore, we select the *89 Rolling - MAPE* scheme for the final numerical study. We also decide to exclude SMA and SES models from our analysis as they have, in general, high MAPE scores.

The final numerical study was done on all the twelve parts using the *89 Rolling - MAPE* scheme. The primary purpose of this study was to determine the forecasting model to be used to generate forecasts for all parts. Tables 3.6 and 3.7 present the average daily MAPE values obtained by running the trained individual and ensemble forecasting models, respectively, to make predictions on the test data for all twelve parts.

Table 3.6: Average of daily MAPE values for individual models (using 89 Rolling - MAPE scheme)

Material number	TES	SARIMA	Prophet
Low runner - 1	60.02	56.30	57.62
Low runner - 2	55.88	52.39	60.32
Low runner - 3	140.47	158.58	133.12
Low runner - 4	90.71	91.23	85.81
Low runner - 5	49.24	45.94	54.73
Low runner - 6	47.22	47.42	57.05
High runner - 1	134.90	108.27	127.25
High runner - 2	52.64	48.40	55.24
High runner - 3	105.92	88.98	80.38
High runner - 4	79.85	73.49	82.90
High runner - 5	92.82	57.80	65.29
High runner - 6	49.07	51.25	68.99
Average	79.90	73.34	77.39
Maximum	140.47	158.58	133.12

Table 3.7: Average daily MAPE values for Ensemble models (using 89 Rolling MAPE scheme)

Material number	SA TS	WA TS	SA TSP	WA TSP	SA SP	WA SP
Low runner - 1	52.29	52.52	51.68	51.66	54.98	55.04
Low runner - 2	54.01	54.37	54.99	55.15	55.54	55.95
Low runner - 3	149.57	149.30	142.92	143.78	144.61	144.13
Low runner - 4	90.90	90.98	87.67	87.71	87.64	87.63
Low runner - 5	47.44	47.46	49.16	49.26	49.57	49.78
Low runner - 6	47.25	47.23	49.56	48.90	51.18	50.11
High runner - 1	115.66	115.72	118.01	119.20	117.64	116.16
High runner - 2	49.55	49.95	50.08	50.27	50.18	50.80
High runner - 3	95.83	96.11	87.02	87.24	82.33	82.13
High runner - 4	76.28	76.57	76.64	76.63	76.14	75.25
High runner - 5	72.56	74.66	67.33	67.82	60.46	61.09
High runner - 6	49.97	49.67	54.91	53.41	59.20	57.89
Average	75.11	75.38	74.16	74.25	74.12	73.83
Maximum	149.57	149.30	142.92	143.78	144.61	144.13

We observe that the MAPE values obtained using the SARIMA model are either best or better than the worst performing individual model except for the parts *Low runner - 3* and *Low runner - 4*. Further, we observe from the values in Table 3.6 and 3.7 that all ensemble models are better than the worst performing individual model for all parts. Therefore, we focus our attention on the ensemble models. To choose the best ensemble model out of the six, we look at the average and maximum values of average daily MAPE per part per week obtained using different ensemble models. Looking at the average values (in the second last row of Table 3.6 and 3.7), we observe that the weighted ensemble model using SARIMA and Prophet performs tantamount to the individual SARIMA model, which has the lowest average MAPE score. Further, the maximum MAPE value (in the last rows of Tables 3.6 and 3.7) obtained using the weighted ensemble model using SARIMA and Prophet is significantly lower than the maximum MAPE value obtained using the individual SARIMA model. Therefore, we propose the weighted ensemble model that use SARIMA and Prophet to generate forecasts for all the parts that will be used to pack the sea-containers using daily volumes instead of the weekly volumes.

To study how well our selected daily forecasting model performs compared to the weekly forecasting model currently used by the manufacturer, we need to aggregate the forecasted daily volumes generated by our model to the weekly level. The manufacturer (i.e., our industrial partner) currently uses a proprietary forecasting model which takes inputs from multiple factors apart from the weekly historical demands. Due to the sensitivity of the information, we did not have access to the details related to the forecasting model currently used by the manufacturer. The results for the weekly average MAPE scores obtained using the selected model against the model currently used by the manufacturer for all parts are presented in Table 3.8.

Table 3.8: Average weekly MAPE values for WA SP and the current model used by the industrial partner

Material number	WA SP	Current BMW forecasting model
Low runner - 1	25.63	12.25
Low runner - 2	16.47	13.80
Low runner - 3	79.41	92.94
Low runner - 4	25.40	97.40
Low runner - 5	11.34	11.12
Low runner - 6	11.96	11.23
High runner - 1	33.48	13.00
High runner - 2	12.27	12.17
High runner - 3	30.50	84.74
High runner - 4	38.76	18.75
High runner - 5	21.94	14.76
High runner - 6	26.93	15.11

The results presented in Table 3.8 show that there is no clear evidence that one model is better in terms of accuracy than the other. However, using the proposed model (the weighted ensemble model using SARIMA and Prophet) has a significant advantage over the current forecasting model used by the manufacturer since our model can forecast daily demand, which the manufacturer's model cannot. The consolidation facility currently packs the weekly order quantity of each part into a single container as a whole because they do not have any information related to the manufacturer's daily demand. Given that the manufacturer's warehouse is designed to hold less than a week's supply (one and a half days supply) of a material, the current consolidation process causes overutilization of the warehouse when a sea-container containing weekly supplies is unloaded. Using our forecasting model, the manufacturer can generate forecasts for the daily demand values, which can help the consolidation facility pack the sea-containers using the daily volumes and thus reduce the manufacturer's warehouse utilization on a daily basis.

3.5 Optimization problem formulation

The reported MAPE values and our discussion in the previous section shows that no forecasting model can be perfectly accurate. Given that the manufacturer places orders with its overseas suppliers using forecasted demand, there is often either excess or shortage of inventory for various parts at the manufacturer's warehouse. As discussed in section 3.3, having excess inventory or shortages in the manufacturing warehouse is undesirable for the manufacturer. However, having shortages are more problematic than having excess inventory because they may cause a production shutdown. Hence, one must avoid shortages as much as possible while reducing the excess inventory in the manufacturing warehouse. Given that we have uncertainty in forecasting errors, we formulate our problem using the chance constraint programming approach because the manufacturer is interested in obtaining the daily supplier emergency safety stock levels in a week that would eliminate shortages in $(1-\varepsilon)\%$ of all possible scenarios. A detailed description of our chance constraint problem is as follows.

Table 3.9: Certain input parameters notation and description

Notation	Description
i	Day index (i.e., 1=Monday, ..., 7=Sunday)
N	Total number of scenarios
n	Scenario index
h	per unit holding cost of excess inventory
b	per unit back-order cost
I_c	Inventory position at the beginning of the week for which we are solving the problem
q_i	Forecast for day i
ε	Significance level

Table 3.10: Uncertain input parameters notation and description

Notation	Description
ξ_i^n	Realization of error in day i's forecast for n^{th} scenario

Table 3.11: Decision variables notation and description

Notation	Description
x_i	Supplier safety stock quantity for day i
y_i^n	Excess or shortage inventory at the end of of day i in the n^{th} scenario
e_i^n	Excess inventory at the end of day i in the n^{th} scenario
s_i^n	Shortages at the end of day i in the n^{th} scenario
w_i^n	Binary variable equals one if there is shortage on day i in the n^{th} scenario
z^n	Binary variable equals one if there is shortage of material on any day in the n^{th} scenario

Chance Constraint Programming Model

$$\text{Minimize:} \quad \sum_{i=1}^{7}\sum_{n=1}^{N} h * e_i^n + \sum_{i=1}^{7}\sum_{n=1}^{N} b * s_i^n \tag{3.1}$$

Subject to:

1. Excess inventory and shortages at the end of each day:

$$\begin{aligned} y_1^n &= x_1 + I_c - (q_1 + \xi_1^n) \quad \ldots \forall n \\ y_i^n &= x_i + y_{i-1}^n - (q_i + \xi_i^n) \quad \ldots i > 2 \ \& \ \forall n \\ e_i^n &\geq y_i^n \quad \ldots \forall i, n \\ s_i^n &\geq -y_i^n \quad \ldots \forall i, n \end{aligned} \tag{3.2}$$

2. '$w_i^n = 1$' if the there is shortage observed on day i in scenario n:

$$\begin{aligned} y_i^n + M * w_i^n &\geq -1 + \varepsilon \quad \ldots \forall i, n \\ s_i^n &\geq w_i^n \quad \ldots \forall i, n \end{aligned} \tag{3.3}$$

3. 'z^n=1' if '$w_i^n = 1$' for any day i in scenario n:

$$
\begin{aligned}
z^n &\geq w_i^n \quad \ldots \forall i, n \\
z^n &\leq \sum_{i=1}^{7} w_i^n \quad \ldots \forall n
\end{aligned}
\tag{3.4}
$$

4. At most εN z^n variables can be equal to one:

$$
\sum_{i=1}^{N} z^n \leq \varepsilon N \tag{3.5}
$$

5. Integer variables:

$$
\begin{aligned}
y_i^n &\in \mathbb{Z} \quad \ldots \forall i, n \\
e_i^n &\in \mathbb{Z}_{\geq 0} \quad \ldots \forall i, n \\
s_i^n &\in \mathbb{Z}_{\geq 0} \quad \ldots \forall i, n \\
x_i &\in \mathbb{Z}_{\geq 0} \quad \ldots \forall i \\
w_i^n &\in \{0,\ 1\} \quad \ldots \forall i, n \\
z^n &\in \{0,\ 1\} \quad \ldots \forall n
\end{aligned}
\tag{3.6}
$$

3.6 Solution methodology and numerical study

We first study the distribution of errors in the daily forecasting model we developed for all twelve parts for the forecast values obtained on testing data (i.e., for the period of thirteen weeks). Our goodness-of-fit analysis done using the JMP software package showed that more than half of the parts follow a mixture of three normal distributions. We randomly selected one high runner and one low runner among these parts to evaluate the chance constraint optimization model we formulated in Section 3.5.

Since the forecast errors we want to incorporate in our model follow a continuous distribution, we resort to approximating the problem using a finite number of scenarios. In particular, we solve our chance constraint formulation via a sampling approach. We refer to the works of Ahmed and Shapiro (2008) and Luedtke and Ahmed (2008) for the sample approximation approach.

To solve a chance constraint optimization problem using the sample approximation approach, we need to determine (1.) the number of times we will solve the problem (i.e., number of replications) to obtain the candidate feasible solutions and (2.) the number of scenarios we solve in each replication. Solving a problem with a higher number of replications (M) and scenarios in each replication is desirable. However, there are limitations on the number of iterations that can be made and the scenarios that can be considered because the computational time required to solve the problem increases with these quantities. We need to solve the problem such that the desired solution quality is achieved in a reasonable amount of time. We use the *Posteriori Feasibility Checking* method described in Luedtke and Ahmed (2008) to obtain the feasible solutions. In this method, we solve a sample approximation problem with a smaller sample size (N) and significance level 'α' to obtain a candidate feasible solution ($\hat{x}$). Another problem is constructed using a sample of size N' (considerably larger than N) and a significance level 'ε' to evaluate the number of scenarios in which shortages are observed using the candidate feasible solution. The significance level 'ε' (nominal risk level) is the maximum desired level of risk the decision-maker is willing to take. The significance level 'α' is used to construct the sample approximation problem, where $\alpha \leq \varepsilon$. Lower is the value of α, the higher is the probability that the optimal solution of the approximation problem is feasible for the nominal risk level problem with N' scenarios.

In our experimental study, we solve the sample approximation problem using (1.) 0.02, 0.03, *and* 0.04 as the significance level α and (2.) 500, 750, *and* 1000 as the number of scenarios (N) in each replication. The decision-maker's desired nominal risk level (ε) is considered 0.04, and a sample of $25,000$ scenarios is used to conduct a posteriori analysis. Further, the number of replication was chosen to be 10.

To obtain the lower bounds for the sample approximation problem with N scenarios, we solve 10 replication of the sample approximation problem with $\alpha = \varepsilon = 0.04$ with N scenarios. Since we solve our problem by considering 10 replications, a lower bound with 0.999 confidence is obtained by taking the lowest value amongst all optimal values, a lower bound with 0.989 confidence is obtained by taking the second-lowest value amongst all the optimal values, and a lower bound with 0.945 confidence is obtained by taking the third-lowest value amongst all the optimal values (Luedtke and Ahmed, 2008). In our problem, the optimality gaps are computed using the lower bounds obtained with 0.945 confidence. Tables 3.12 and 3.13 present the average optimality gap values obtained over the period of thirteen weeks for different experimental setups for high runner and low runner, respectively.

Table 3.12: Optimality gap for the high runner part with at least 94.5% confidence

Scenarios	500		750		1000	
Statistics	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation
$\alpha = 0.02$	9.12%	1.55%	8.00%	1.41%	7.23%	1.39%
$\alpha = 0.03$	***	***	7.04%	1.42%	5.72%	1.37%
$\alpha = 0.04$	***	***	***	***	***	***

Table 3.13: Optimality gap for the low runner part with at least 94.5% confidence

Scenarios	500		750		1000	
Statistics	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation
$\alpha = 0.02$	10.22%	0.80%	8.23%	0.99%	7.87%	1.20%
$\alpha = 0.03$	***	***	7.85%	1.20%	6.92%	1.31%
$\alpha = 0.04$	***	***	***	***	***	***

From the values presented in Tables 3.12 and 3.13 we observe that the lower optimality gaps can be obtained by increasing the number of scenarios and the significance level α. However, increasing one or both of them causes an increase in the computational time required to solve the problem.

Next, we discuss the quality of the candidate feasible solutions and the number of feasible solutions generated using the sample approximation approach. We obtain M candidate solutions, each corresponding to daily demand quantities in a given week, by running the sample approximation problem with N scenarios and the significance level α for M replications. We then solve the nominal problem with N' scenarios M times, each time by using the daily demand quantities (in a given week) obtained by solving the sample approximation problem with N scenarios and the significance level α for M replications. For each solution, we compute the solution risk value defined as the percentage of infeasible scenarios (i.e., the number of scenarios in which we observe shortages) observed in the nominal problem. We report the average solution risk value observed each week for high runner and low runner parts in Figures 3.3 and 3.4, respectively.

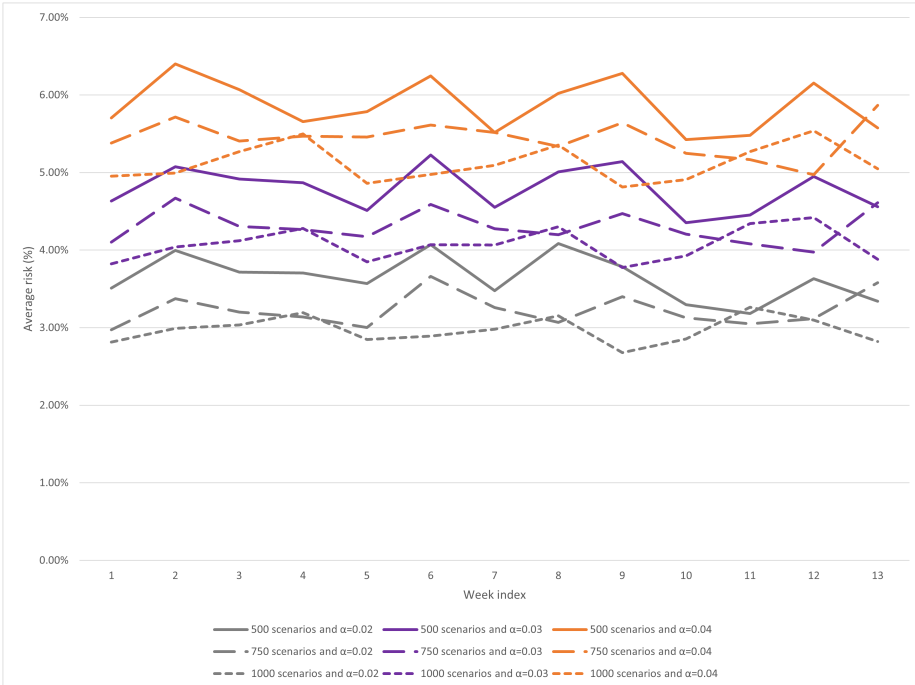


Figure 3.3: Average solution risk observed each week for the high runner part

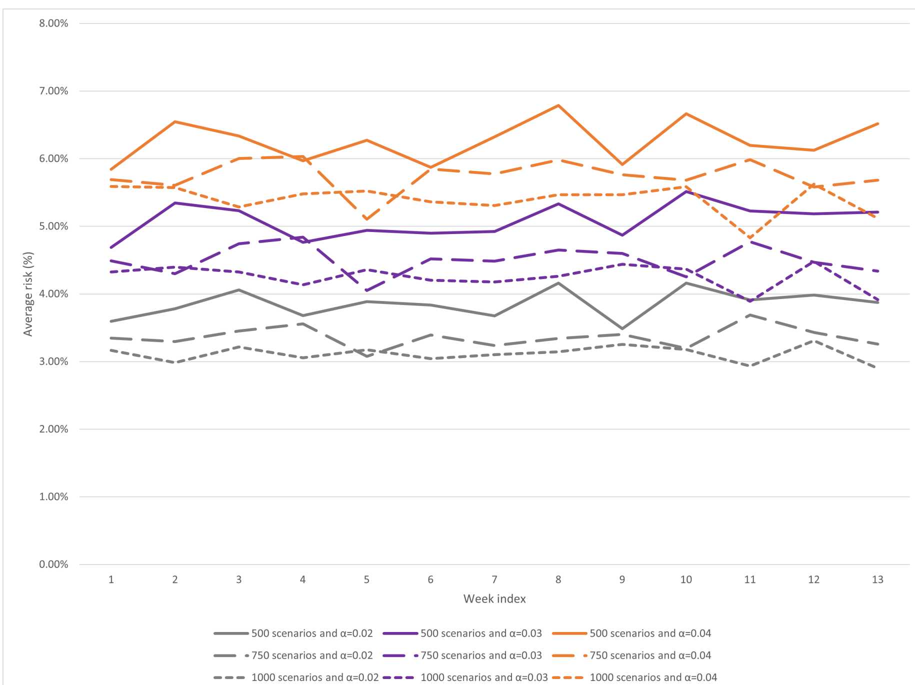


Figure 3.4: Average solution risk observed each week for the low runner part

By analyzing the plots in Figures 3.3 and 3.4 for experiments run using different number of scenarios (N) and significance levels (α), we observe that the average solution risk value observed in the nominal problem increases as we increase the value of significance level α. Further, we observe that the average solution risk value observed in the nominal problem increases as we decrease the number of scenarios (N) solved in each sample approximation problem. A lower solution risk is desirable because it implies a high likelihood of obtaining a feasible solution to the nominal problem.

In Tables 3.14 and 3.15, we provide the average number of feasible solutions obtained per week corresponding to different parameter settings in sample approximation problem for high runner and low runner parts, respectively. Further, in Figures 3.5 and 3.6 we plot the number of feasible solutions obtained each week for high runner and low runner parts, respectively.

Table 3.14: Average number of feasible solutions obtained each week for the high runner part

Scenarios	500	750	1000
$\alpha = 0.02$	7.31	8.62	9.77
$\alpha = 0.03$	1.85	3.54	4.69
$\alpha = 0.04$	0.15	0.15	0.31

Table 3.15: Average number of feasible solutions obtained each week for the low runner part

Scenarios	500	750	1000
$\alpha = 0.02$	5.92	8.38	9.15
$\alpha = 0.03$	1.62	3.00	3.62
$\alpha = 0.04$	0.08	0.00	0.23

Results reported in Tables 3.14 and 3.15, show that as we decrease the value of significance level α, we obtain more feasible solutions to the nominal problem. Further, we also observe that as we increase the number of scenarios (N), we obtain more feasible solutions to the nominal problem. A similar observation can also be based on Figures 3.5 and 3.6.

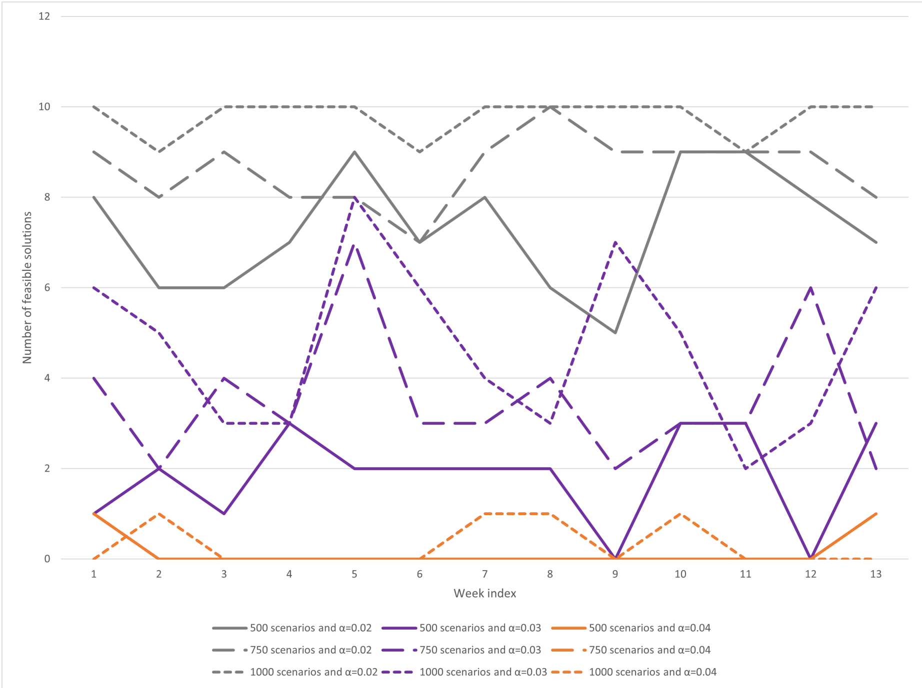


Figure 3.5: Number of feasible solutions obtained every week for the high runner part

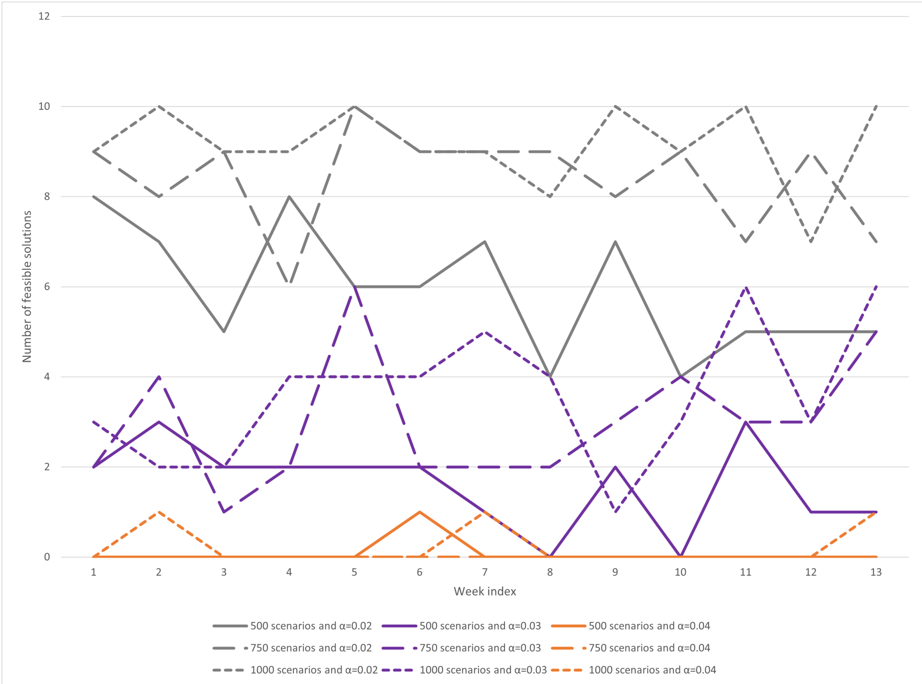


Figure 3.6: Number of feasible solutions obtained every week for the low runner part

In Tables 3.16 and 3.17, the weekly excess emergency stock quantity at the supplier is presented for high runner and low runner parts, assuming that the manufacturer uses the solution obtained by solving the chance constraint optimization model to determine the supplier's emergency safety stock level for a given week.

Table 3.16: Weekly excess emergency stock at the supplier for the high runner part

Week index	500 scenarios & α =0.02	750 scenarios & α =0.02	750 scenarios & α =0.03	1000 scenarios & α =0.02	1000 scenarios & α =0.03
1	3301	2795	2903	3131	2968
2	5195	5137	4888	5549	5117
3	5350	5193	4933	4857	5393
4	4233	4362	4185	4170	4003
5	5002	4474	4528	4810	4820
6	4763	4152	4035	4391	3989
7	3247	2987	3419	3524	3235
8	5531	5395	5171	5236	5662
9	3202	3088	3300	3404	3226
10	5309	6005	5271	5514	5602
11	3009	3383	2892	3587	3196
12	5049	5201	5478	5233	5352
13	5377	5185	5247	5200	5838

Table 3.17: Weekly excess emergency stock at the supplier for the low runner part

Week index	500 scenarios & α =0.02	750 scenarios & α =0.02	750 scenarios & α =0.03	1000 scenarios & α =0.02	1000 scenarios & α =0.03
1	1053	1007	1016	998	895
2	1064	1138	998	1036	924
3	1151	1022	972	1052	951
4	1054	950	1085	1164	1038
5	1050	1111	973	1031	1019
6	1015	957	997	1043	1003
7	908	934	871	984	893
8	1117	1088	1073	992	1003
9	534	444	461	314	403
10	1037	944	1088	1034	1031
11	513	477	451	517	502
12	895	830	961	869	914
13	1146	1001	995	1087	973

The results reported in Tables 3.12 – 3.15, show that solving a sample approximation problem with a higher number of scenarios is desirable because we obtain lower optimality gaps and a higher number of feasible solutions. We also observe that a larger number of feasible solutions are generated with lower values of α.

By comparing the values of excess emergency inventory at the end of each week obtained by solving the sample approximation problem with 750 and 1000 scenarios and a significance level of 0.02 (i.e., see Tables 3.16 and 3.17), we observe that increasing the number of scenarios considered from 750 to 1000 does not result in a significantly lower amount of excess emergency inventory. Therefore, we propose that the manufacturer use and solve the sample approximation problem with 750 scenarios and a significance level of 0.02.

Generally, the supplier maintains two weeks of stock at the supply hub, and the manufacturer pays for the material consumed by it (Barnes et al., 2003). However, in our problem, the industrial partner requires the suppliers to maintain 25% more stock than the forecasted value, and the manufacturer pays for the material it consumes. Therefore, we compare three different approaches to determine the supplier inventory levels, which are as follows: (1.) Supplier inventory level using chance constraint optimization model, (2.) Supplier inventory level using two weeks of forecasted demand, and (3.) Supplier inventory level using 1.25 times the forecasted demand. In Figures 3.7 and 3.8, we plot the excess emergency stock levels at the supplier observed at the end of each week using different approaches, and the weekly shortage quantities (shortages as a result of using the order quantities obtained from the daily forecasting models) observed by the manufacturer. The values for excess emergency stock levels at the supplier plotted in Figures 3.7 and 3.8 were computed as follows:

- Excess supplier emergency stock using chance constraint model = Weekly chance constraint demand - forecasted demand - emergency orders
- Excess supplier emergency stock using 2 weeks of forecasted = 2*Weekly forecasted demand - forecasted demand - emergency orders
- Excess supplier emergency stock using 25% of forecasted = 1.25*Weekly forecasted demand - forecasted demand - emergency orders

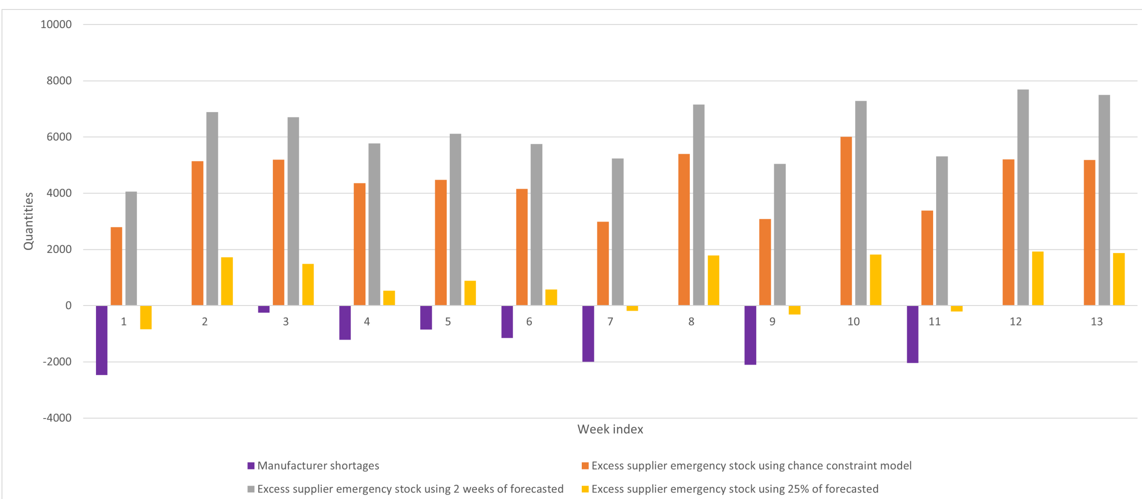


Figure 3.7: Weekly excess emergency stock at the supplier and shortages at manufacturer for the high runner part

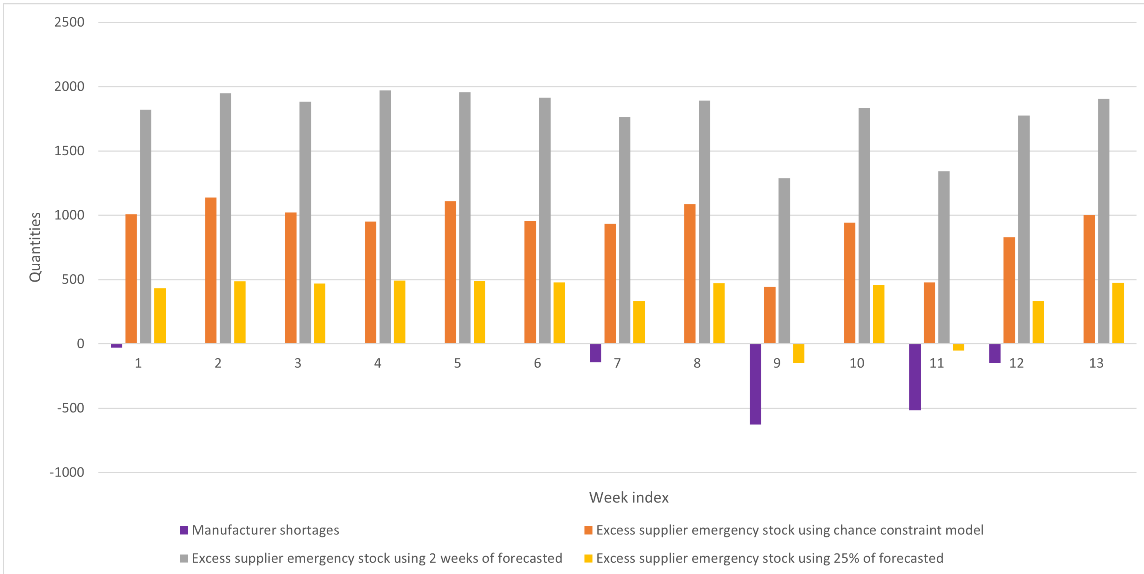


Figure 3.8: Weekly excess emergency stock at the supplier and shortages at manufacturer for the low runner part

In Figures 3.7 and 3.8, the manufacturer shortage values reported are the shortages faced as a result of using only the inventory of materials procured using the forecasted daily demands. These shortages are mitigated by placing emergency air-freight orders which are fulfilled using the emergency stock levels maintained at the suppliers. The excess supplier inventory reported in these figures is the amount of stock remaining at the supplier after fulfilling any emergency orders. Higher excess emergency stock levels at the end of each week are observed by maintaining the stock levels using *two weeks of forecasted demand* approach. In contrast, shortages are observed at the suppliers for multiple weeks if the stock levels are maintained using the currently practiced approach by the industrial partner. The solution obtained using the proposed chance constraint optimization model results in maintaining a sufficient amount of safety stock with suppliers for all weeks such that neither excessive overstocks (as in the case of *two weeks of forecasted demand* approach) nor shortages (as in the case of currently practiced approach by the industrial partner) are observed with suppliers.

In Figures 3.9 and 3.10, we present the plots for excess inventory in the manufacturer's warehouse at the end of each day for seventy seven consecutive production days. In the case of weekly consolidation, we assume that the entire week's order for a part is consolidated into a single sea-container, and this container is unloaded at the beginning of the consumption week to avoid detention penalty costs. Similarly, in the case of daily consolidation, we assume that the single day's demand for a part is consolidated into a sea-container. This container can be unloaded on any day corresponding to the week for which it was built and all the containers belonging to a given week need to be unloaded by the end of the week. The manufacturer uses such a consolidation process only for parts with weekly demands that can be packed into a single container. However, especially in the case of daily consolidation, the manufacturer would consolidate daily order quantities of multiple parts in the same container to efficiently use container space and reduce transportation costs. This means, in reality, the estimated daily order quantities may not be always used as is for all parts and, in order to obtain an accurate estimate of the daily warehouse utilization, a packing problem under forecast uncertainty needs to be formulated and solved. However, the values presented in Figures 3.9 and 3.10, provide an estimate on optimistic warehouse utilization reduction.

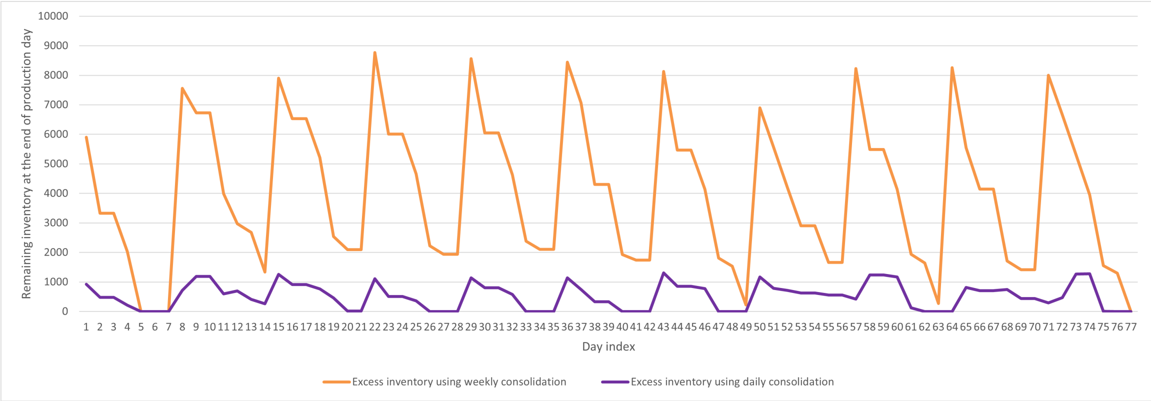


Figure 3.9: Comparing excess inventory of the high runner part at the end of each day

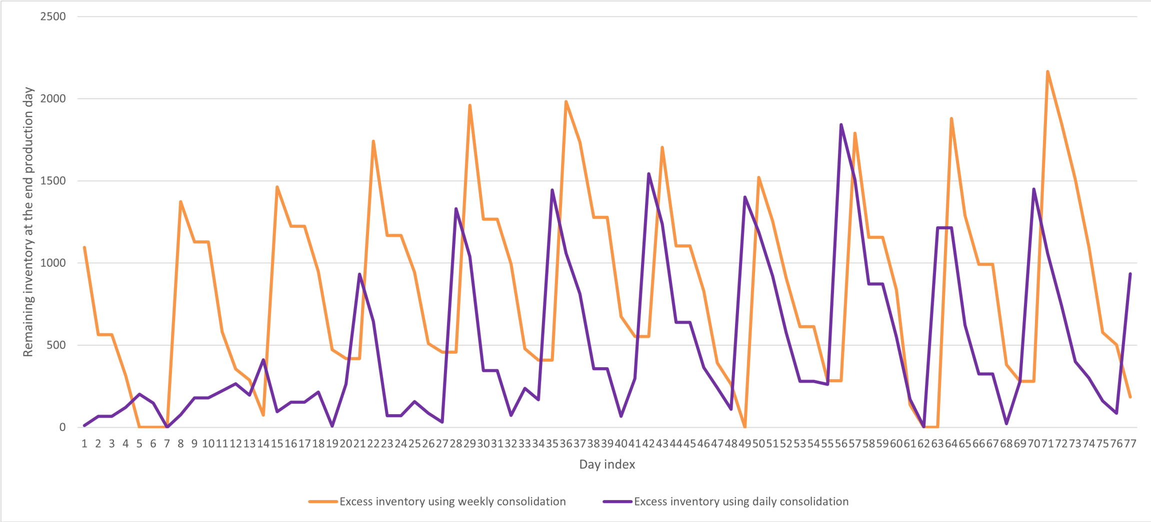


Figure 3.10: Comparing excess inventory of the low runner part at the end of each day

In Figures 3.9 and 3.10, we observe that in the case of high runner part, a higher reduction in daily excess leftover inventory can be achieved using daily consolidation compared to the low runner part. For the high runner part, over a period of seventy-seven consecutive production days, an average reduction of 86.91% is achieved by consolidating the orders using daily forecasted demand instead of weekly. Similarly, in the case of low runner part, an average reduction of 42.04% is achieved. These reductions are optimistic estimates for the actual reductions due to the reasons we discussed in previous paragraphs.

3.7 Conclusions

In this chapter, we built time-series models using various univariate methods, such as Simple Moving Averages, Simple Exponential Smoothing, Holt-Winter's (i.e., Triple Exponential Smoothing), SARIMA, and Prophet. Further, we did a comparative analysis for all these methods using their respective MAPE scores observed on the testing dataset for different parts. Based on the results, we concluded that using a single of these methods is not advisable for generating forecasts for all parts. In order to build a single model that would perform well for all parts, we built an ensemble-based forecasting model combining the forecast values from different forecasting methods to yield the final forecast values. Six different ensemble-based models were developed and thoroughly analyzed using the MAPE scores as a performance metric. We also studied the effects of training data size and the metric (i.e., MAPE and RMSE) used for calculating the weights (for the weighted average ensemble model) on the forecast quality. Our results, showed that we could obtain a good quality of forecasted values by training the models using 89 weeks of historical data. We also observed that for the models trained using 89 weeks of historical data, both MAPE and RMSE metrics perform equally well for computing the weights for the weighted average ensemble model. Further, using 89 weeks of data instead of the entire dataset reduces the computational time required to train the forecasting models. Therefore, we used the 89 weeks of historical data to train our time-series models for different parts. In the case of the weighted average ensemble model, we computed the weights using MAPE values obtained on the training dataset. Based on a numerical study, we concluded that the weighted average ensemble model using the SARIMA and Prophet models performs reasonably well for all parts. We also observed that, in general, the MAPE scores obtained for this daily forecasting model are as good as the industrial partners' weekly forecasting model. An optimistic estimation of the excess inventory suggests a significant reduction in the warehouse utilizations if the container are consolidated using the daily demand instead of the weekly demand values.

We studied the distribution of forecasting errors for all parts and selected two parts, one high runner and one low runner, for further study. We built a chance-constraint optimization model using the uncertain forecasting errors to help the decision-maker calculate the daily safety stock levels at the supplier for any emergency orders. We solved the chance-constraint optimization model using the sample approximation approach. We also did an exhaustive analysis to select the number of scenarios and significance level for the sample approximation problem. The results suggest that by using the lower significance level, a higher number of feasible solutions can be generated with respect to the nominal problem. However, we also observed that the optimality gap increases by reducing the significance level of the sample approximation problem. We conclude that the decision-maker should solve the sample approximation problem by considering the number of scenarios and the significance level that provides the best trade-off between the number of feasible solutions and the optimality gap obtained. The results obtained from the study indicate that the stock-outs are prevented if the supplier safety stock level is maintained based on the chance-constraint solution.

3.8 Potential future research

In this study, the consolidation quantities are calculated using the daily forecasted values. However, based on forecasting results presented in Tables 3.8, we observe that sometimes the weekly forecasts perform better at predicting the weekly volumes. Hierarchical reconciliation-based forecasting can be built, and the forecasted values at the daily and weekly levels can be reconciled by calculating the historical error variance of the forecasted values. An optimal method to reconcile a hierarchical time series using the error variance is given in Wickramasuriya et al. (2019) and can be used for reconciliation such that the overall forecast accuracy is improved.

In Section 3.3, we assume that the order quantities placed by the manufacturer can be adjusted at the time of consolidation. However, if a manufacturer cannot readjust the order quantities with the supplier once they are placed, then either of the following two approaches can be used: (1.) The proposed hierarchical model by Wickramasuriya et al. (2019) can be used by setting the error variance equal to zero for the weekly forecast used to place the orders, and the daily volumes can be readjusted using their respective error variance, or (2.) The ratio for each day's daily demand can be calculated by dividing the daily demand by the sum of the daily demands in a given week. Consequently, the readjusted daily demand can be obtained by multiplying the weekly forecasted demands with the respective daily demand ratios. These two methods can be compared and studied in future research.

In this study, we only focused on computing the daily demand volumes for consolidation and did not solve any vehicle routing or consolidation problems to move the material from inbound trailers into the outbound trailers/sea-containers. To the best of our knowledge, there are no studies that considered the end warehouse utilization while solving the vehicle routing or consolidation problems. The existing studies account for the inventory holding cost and backorder cost for minimizing the supply chain costs. However, these costs are not necessarily proportional to the volume of space the parts occupy in the warehouse. Accounting for the warehouse utilization in terms of bin or volumetric occupancy can potentially improve the overall warehouse performance and reduce the warehouse space requirement due to the reduction in warehouse utilization. Future research can also consider this new perspective of accounting for the end warehouse utilization while solving the vehicle routing and consolidation problems.

Chapter 4

Optimal Containers Selection with Potentially Conflicting Objectives

4.1 Introduction

In this chapter, we investigate warehouse replenishment strategies that balance two potentially conflicting objectives. Managing inventory replenishment to a warehouse is challenging when production plans are subject to frequent changes. The warehouse we consider contains multi-product inventory, where the materials are continuously consumed based on the line-side demand. The sea-containers and trailers carrying materials arrive at the yard (loaded sea-containers and trailers holding facility at manufacturing plant) over time, and detention penalties are incurred on a daily basis for holding them. The warehouse inventory is replenished using the available trailers and sea-containers from the yard. The trailers and sea-containers need to be entirely unloaded when called upon for unloading.

Figure 4.1 depicts the information flow and the decision-making process. The blue lines depict the information flow into the Enterprise Resource Planning (ERP) system through Electronic Data Integration (EDI). The data on the predicted incoming trailers and sea-cans (jointly termed as containers) is available for the next two production days, along with the information for materials held in each of them, as shown in the figure. The data for the containers currently available for unloading and the information for material held in each of them is also available. The delivery date for each container to the manufacturing plant is known, using which we calculate their detention days. The current warehouse stock for different materials is known, along with their daily deterministic demands for today and the next two days is also available. The cloud in Figure 4.1 represents the decision-making done by the ERP system to select the containers for warehouse stock replenishment.

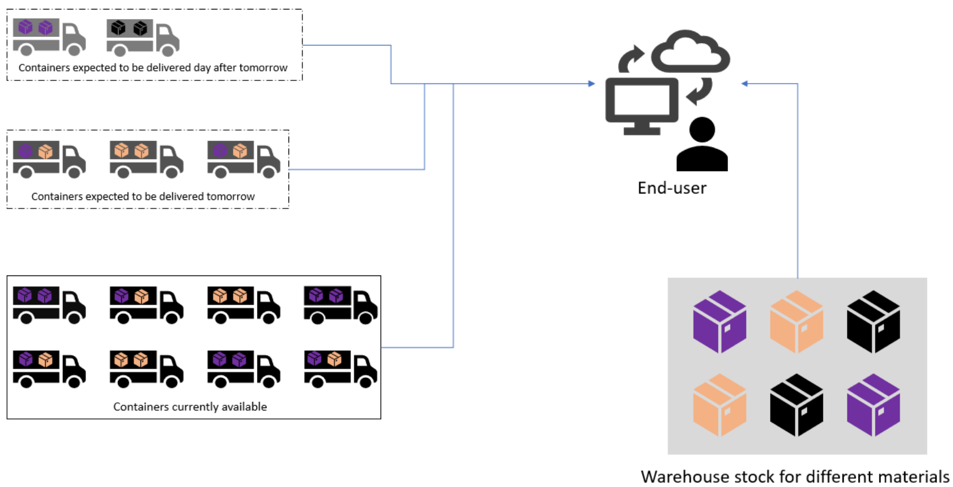


Figure 4.1: Warehouse and yard operations for warehouse stock replenishment

We study the problem of optimally selecting the sea-containers and trailers to unload, to reduce the number of remnant materials in the warehouse, and to minimize the detention penalties. We formulate this problem using a multi-objective integer program and solve it using goal programming and ε-constraint methods. The model is solved over a rolling horizon of three days. Solving a rolling horizon model allows us to predict the future unloads and feasibility of the problem over a longer horizon.

The next section provides a literature review on this problem. We then describe the problem formulation using a multi-objective integer programming model in *Section 4.3*. In *Section 4.4* we discuss the solution strategies. The experimental setup for the preliminary analysis is described in *Section 4.5*, whereas the data requirement and results for preliminary analysis are provided in *Section 4.6* and *Section 4.7* respectively. The observations for the models obtained via results of the preliminary analysis are discussed in *Section 4.8*. In *Section 4.9* we discuss the solution methodology selected for implementation in the production system of the industry we collaborated with, the long-term performance of the selected method is also discussed in this section. We also provide a detailed discussion about observations and conclusions for the solution of experimental models. Finally, the impact value and significance of the problem solved in this chapter are provided in *Section 4.10*.

4.2 Literature review

We focus on a joint yard and warehouse optimization problem. Most of the literature in this area is focused on: (i.) the dock door assignment problems for a cross-dock facility, (ii.) product-allocation inside the warehouse, (iii.) resource-allocation in the warehouse, and (iv.) ordering policies. Here we focus on the studies from the first three veins of literature as we are interested in optimizing the supply chain operations after orders are placed.

The problem for assigning containers/trailers to docks is studied widely in the receiving and shipping area for optimizing the cross-dock operations (Gu et al., 2007). Various objectives are considered for this problem, such as minimizing total operational cost, minimizing total waiting time, and minimizing congestion subject to a set of operational constraints. The warehouse storage problems are also discussed in the literature. However, all the studies discussed in this paper for warehouse storage problems mainly focus on lot sizing, ordering frequency, and/or storage allocation decisions. None of the studies mentioned here in this review considers containers/trailers selection problems to optimize warehouse utilization and containers/trailers tardiness simultaneously. Tsui and Chang (1992) study a problem of dock-door assignment problem, in which the objective is to assign the receiving doors to origin doors so as to reduce the total distance traveled by forklift. Another truck-door assignment problem for inbound and outbound trailers in a cross-dock facility was studied by Miao et al. (2009). Their objective is to minimize the operational and penalty cost for unfulfilled shipments. In this problem, the penalty cost is incurred if any of the trucks is missed in the dock assignment solution. A system of multiple cross-docks is studied by Chen et al. (2006). The goal of this study is to minimize the transportation and inventory handling costs by optimally assigning the inbound and outbound trailers to the set of available cross-dock facilities.

Another set of problems comprises of optimal product allocation inside the warehouse or cross-dock facility. Poulos et al. (2001), study the warehouse replenishment process. They formulate a multi-objective optimization model to optimally assign the storage locations to the set of various products to address the problem. The problem of optimally assigning the products to a storage aisle to reduce the average order picking time in the warehouse was studied by Kutzelnigg (2011). A cross-dock facility wherein a product can be transported from a shipping area to a receiving area using one of the various kinds of flow processes was studied by Heragu et al. (2005). The problem is to assign flow processes to all the products such that the total handling and storage cost in the facility is minimized. The problems of optimizing the pick-up and delivery process from warehouse to line-side and resource allocation is studied by Poon et al. (2011).

A hierarchical optimization model was formulated by Berghman and Leus (2015) to optimally schedule the timing of the tasks involved in the dock operations of the distribution warehouse. The primary objective of their problem is to minimize the weighted sum of the number of late outbound trailers and their tardiness. The secondary objective is to minimize the weighted completion time of all trailers. Optimizing the primary objective is more critical than the secondary objective; therefore, they develop a hierarchical optimization model. In contrast, the optimization of both objective functions is equally important in our problem.

Muriel and Ruiz-Benitez (2006) studied an Integer Programming problem for yard and warehouse management. Their problem is to minimize the overall cost in the warehouse by assigning each inbound trailer to an outbound trailer for the time-period 't' and deciding on the product storage policy in the inbound trailers. The authors propose that the analysis can be restricted to three possible policies called full-truckload, ready-to-go, and partial truck-load to search for an optimal solution. In this problem, all the inbound trailers are considered to have a single product inside them, and the outbound trailers to customers have a mixture of products in them. They assume that the on-hand inventory of the product is always more than the demand for all the products for all time-periods, which in the general case is not always true. They also assume that the decision-maker is more concerned about the total worth of inventory held in the warehouse and the yard, and the handling costs. In our study, the decision-maker is more interested in minimizing the volumetric warehouse occupancy and the detention-penalties incurred by storing the trailers and sea-cans in the yard. Furthermore, our problem is subjected to a different set of operational constraints.

To the best of our knowledge, none of the studies done so far address the problem of simultaneously minimizing the volumetric warehouse utilization and detention penalties of the trailers and sea-cans stored in the yard along with the operational constraints of the system.

4.3 Optimization model development

We develop a model that minimizes the amount of remnant parts in the warehouse and the detention penalties incurred for holding containers for a long time. These two objectives are represented by equations (4.1) and (4.2) given under *General Multi-Objective Mixed Integer Program* in this section. To minimize the detention penalties incurred on the containers that remain in the yard each day, we maximize the detention penalties associated with the containers selected for unloading in one of the objective functions. Thus, the model first tries to unload the older containers (i.e., the ones with higher penalties). Further, we are also interested in predictions on the set of containers to be unloaded in the next two production days. Thus, we solve the model over a three-day period on each day using a rolling horizon. The new containers and trailers loaded with materials arrive daily at the plant. Therefore, we use the Advanced Shipment Notification (ASN) data to identify the trailers and containers expected to be delivered within the next two days.

A chosen container needs to be unloaded either on the current day or is predicted to be unloaded on one of the next two days. The binary decision variables determine the set of containers to be unloaded on each day. The selected containers should also meet specific requirements on daily demand fulfillment, warehouse utilization, and containers and trailers unload capacity. These constraints are discussed below.

Ideally, the daily stock requirements should be met for all parts. However, the plant often has a higher stock requirement than the cumulative available stock on all the containers of a given part. If this is the case, the demand needs to be partially met as much as possible by unloading all the containers that contain such parts. The constraints given in equation (4.3) ensure that the daily stock requirements for all the parts are met either entirely or partially (if enough cumulative stock is not available on all containers). The constraints given in equation (4.6) ensure that the daily warehouse utilization is maintained below a particular percentage value on all production days. Maintaining the daily warehouse utilization below specified limits is essential to ensure smooth warehouse operations. Further, there is also a capacity constraint on the maximum number of containers that can be unloaded daily due to the limited workforce availability and warehouse space. The constraint given in equation (4.7) ensures that the number of containers and trailers unloaded on a particular day is not more than the permissible number. The constraint given in equation (4.8) ensures that the containers expected to be delivered later in the three-day horizon are not chosen by the model to be unloaded before their delivery.

We formulated this problem using a bi-objective optimization model with two potentially conflicting objectives. The first objective is to minimize the number of remnant parts in the warehouse. The second objective is to maximize the detention penalties associated with the selected sea-containers and trailers. These objectives may conflict because the second objective may lead to excluding containers with a favorable load-mix due to lower detention days. Thus, resulting in unloading more containers to meet the stock requirement and flooding the warehouse with excessive remnant parts.

Table 4.1: Notations used in multi-objective optimization model

Sets	
A	Set of all the parts
I	Set of required parts ($I \subseteq A$)
N	Set of all the available sea-containers and trailers
N_j	Set of all the sea-containers and trailers available on day j ($N_0 \subseteq N_1 \subseteq N_2$)
N_S	Set of all the available sea-containers ($N_S \subseteq N$)
N_T	Set of all the available trailers ($N_T \subseteq N$)
a	Element of set A
n	Element of set N
Parameters	
j	0, 1 or 2 (0: today, 1: tomorrow, 2: day after tomorrow)
$d_{n,j}$	Number of days the container n spent in the storage until day j
$r_{a,j}$	Quantity of part a required to satisfy demand on j^{th} day
	• $r_{a,j} > 0 \ldots$ if $a \in I$
	• $r_{a,j} = 0 \ldots$ if $a \notin I$
$S_{a,n}$	Quantity of part a in container n
V_a	Volume of part a
w_a^0	Quantity of part a in the warehouse at $j = 0 (today)$
C	Holding cost of a container per day
T	Total available space in warehouse
μ	Threshold percentage value for maximum warehouse utilization
α_S	Maximum number of sea-containers that can be unloaded on a given day
α_T	Maximum number of trailers that can be unloaded on a given day
Decision variables	
$x_{n,j}$	Binary decision variable which takes value 1
	when the container n is selected for unloading on day j
$w_{a,j}$	Quantity of part a in the warehouse on day $j > 0$

General Multi-Objective Mixed Integer Program

$$\text{minimize} \quad f_1(x_{n,j}) = \sum_{a\in A}[[\sum_{j=0}^{2}[\sum_{n\in N}(x_{n,j} * S_{a,n}) - r_{a,j}]] + w_a^0] \tag{4.1}$$

$$\text{maximize} \quad f_2(x_{n,j}) = \sum_{j=0}^{2}\sum_{n\in N} C * d_{n,j} * x_{n,j} \tag{4.2}$$

Subject to:

1. Satisfying the daily demand for part a on j^{th} day:

$$\left.\begin{aligned} \sum_{n\in N} x_{n,0} * S_{a,n} \geq min\{r_{a,0} - w_a^0, \sum_{n\in N} S_{a,n}\} \\ \sum_{n\in N} x_{n,j} * S_{a,n} \geq min\{r_{a,j} - w_{a,j}, \sum_{n\in N} S_{a,n}\} \end{aligned}\right\} \quad \dots (\forall\ a) \tag{4.3}$$

2. Amount of stock for part a in warehouse on $j = 1$:

$$w_{a,1} = \sum_{n\in N} x_{n,0} * S_{a,n} - r_{a,0} + w_a^0 \quad \dots (\forall\ a) \tag{4.4}$$

3. Amount of stock for part a in warehouse on $j = 2$:

$$w_{a,2} = \sum_{n\in N} x_{n,1} * S_{a,n} - r_{a,1} + w_{a,1} \quad \dots (\forall\ a) \tag{4.5}$$

4. Maintaining the warehouse utilization below permissible limit:

$$\left.\begin{aligned} \frac{\sum_{a\in A} V_a[\sum_{n\in N}(x_{n,0} * S_{a,n}) + w_a^0 - r_{a,0}]}{T} \leq \mu \\ \frac{\sum_{a\in A} V_a[\sum_{n\in N}(x_{n,j} * S_{a,n}) + w_{a,j} - r_{a,j}]}{T} \leq \mu \end{aligned}\right\} \quad \dots (j > 0) \tag{4.6}$$

5. Maximum sea-can and trailer unload capacity:

$$\left.\begin{aligned} \sum_{n\in N_S} x_{n,j} \leq \alpha_S \\ \sum_{n\in N_T} x_{n,j} \leq \alpha_T \end{aligned}\right\} \quad \dots (\forall j) \tag{4.7}$$

6. Containers unavailable for unloading on day j:

$$\left.\begin{aligned} x_{n,0} = 0 \quad & \ldots (\forall\, n \notin N_0) \\ x_{n,1} = 0 \quad & \ldots (\forall\, n \notin N_1) \end{aligned}\right\} \tag{4.8}$$

7. Variable value restrictions:

$$\left.\begin{aligned} x_{n,j} \in \{0,\ 1\} \quad & \ldots (\forall\, n,\ j) \\ w_{a,j} \in \mathbb{Z}_{\geq 0} \quad & \ldots (\forall\, a \text{ and } j > 0) \end{aligned}\right\} \tag{4.9}$$

4.4 Solution strategies

There are multiple ways to solve multi-objective optimization problems: the weighted-sum method, lexicographic method, ε-constraint method, goal programming method and metaheuristic approaches. In our problem, we cannot quantify the weights of each objective dependably. Hence we cannot use the weighted-sum method. In the lexicographic method, the decision-maker needs to provide the ranking of the objectives. The optimization is done individually on each objective using these rankings. After optimizing over the highest ranked (most important) objective, if only one unique solution is obtained, then the optimization stops. Whereas, if multiple solutions are obtained, then we continue with the optimization using the second-highest ranked objective and a constraint on the solution obtained from the previous objective. Our industry partner views the optimization of the two objectives as equally important, thus the lexicographic method is not suitable for a practical implementation. In the ε-constraint method, we optimize by considering one objective function and converting another objective function into a constraint. The optimization model is then solved for different right-hand-side (RHS) values of the objective converted into constraint, thus generating multiple Pareto optimal solutions. When using the goal programming method, the decision-maker needs to provide the aspired target values for each objective function. The goal programming method minimizes the sum of squared errors between the obtained objective values and decision-maker defined target values. Metaheuristic algorithms have been used to generate a near-optimal solution for complex and difficult-to-solve problems. Ant colony optimization, simulated annealing, and genetic algorithms are a few examples of metaheuristic algorithms. Ant colony optimization and simulated annealing algorithm have been used to solve scheduling, vehicle routing, and traveling salesman problems. NSGA-II, a genetic algorithm, has been used to solve multi-objective optimization problems. In this study, we used goal programming (G.P.), the ε-constraint (ε-con) and NSGA-II method as our solution strategies.

To reformulate the bi-objective optimization problem using the ε-constraint method, we select the minimization of remnant parts as the objective function and modify the other objective function (i.e., maximization of detention costs for selected containers) into a constraint. This constraint is formulated to impose targeted minimum average detention days for the selected containers, mathematically represented by equation (4.11). The problem is then solved for different values of minimum average detention days that need to be achieved, represented by 'ε' in equation (4.11). While using this method, one should determine the lower and upper bound on 'ε' values to be considered. In our case, the lower bound is always zero as the number of detention days cannot be negative. The upper bound is determined by the container with the highest detention days and can be looked up in the available container data. The mathematical model for this solution strategy is presented under *ε-constraint model* in this section.

ε-constraint model

$$\text{minimize} \quad f_1(x_{n,j}) = \sum_{a\in A}[[\sum_{j=0}^{2}[\sum_{n\in N}(x_{n,j} * S_{a,n}) - r_{a,j}]] + w_a^0] \tag{4.10}$$

Subject to: (4.3) - (4.9)

$$\sum_{\forall n}\sum_{\forall j} x_{n,j} * d_{n,j} \leq \sum_{\forall n}\sum_{\forall j} x_{n,j} * \varepsilon \tag{4.11}$$

The goal programming method minimizes the sum of squared errors between the obtained objective values and utopia point values and produces a solution closest to the target objective function values. In our model, the minimum number of remnant parts that can be achieved is zero (target value for the first objective function). We calculate the target value for the second objective function using the available detention days data. We select the first 'N' containers and sum their current detention days in the yard. The number 'N' is obtained by summing the daily unloading capacity for the three days in the rolling horizon. The mathematical model for this solution strategy is presented under *Goal programming model* in this section.

Goal programming model

$$\text{minimize} \quad (\sum_{a\in A}[[\sum_{j=0}^{2}[\sum_{n\in N}(x_{n,j} * S_{a,n}) - r_{a,j}]] + w_a^0] - 0)^2 + (\sum_{j=0}^{2}\sum_{n\in N} C * d_{n,j} * x_{n,j} - \delta)^2 \tag{4.12}$$

Subject to: (4.3) - (4.9)

Multi-objective optimization problems can also be solved using evolutionary algorithms. To solve the multi-objective optimization problem discussed in this chapter, we use the NSGA-II method developed by Deb et al. (2002). Figure 4.2 shows the flowchart for the NSGA-II algorithm. A description of the steps involved in implementing the NSGA-II is given below.

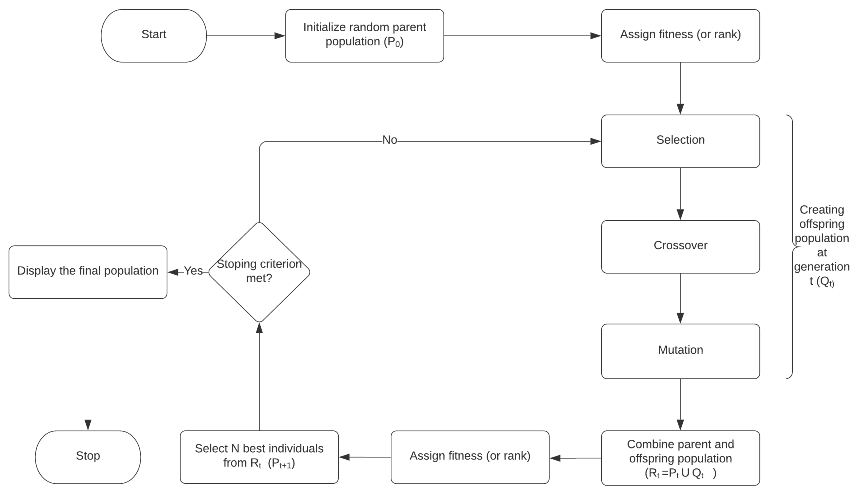


Figure 4.2: NSGA-II flow-chart

NSGA-II

Step 0 - Initialization: Create a random initial population (P_0) of size N. This population can contain both feasible and infeasible solutions.

Step 1 - Assigning a fitness (or rank): Find the non-dominated fronts ($\mathscr{F}_1$, $\mathscr{F}_2, \ldots, \mathscr{F}_n$) using the population. An example of the non-dominated fronts is shown in Figure 4.3. The figure represents the non-dominated fronts for a problem with two minimization objectives $g_1(x)$ and $g_2(x)$. Therefore, in Figure 4.3 the solutions belonging to front $\mathscr{F}_1$ are most preferable and the solutions belonging to front $\mathscr{F}_3$ are least preferable. The non-dominated sorting approach described in Deb et al. (2002) is used to find these non-dominated fronts.

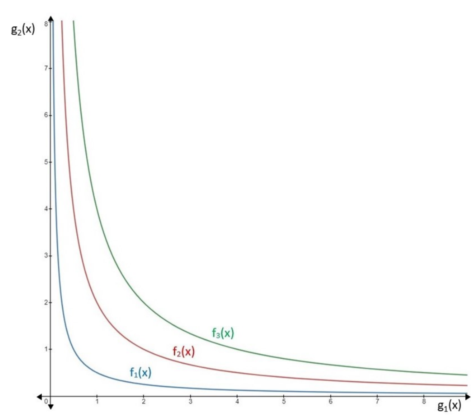


Figure 4.3: Non-dominated fronts

Step 2 to *4* describe the creation of the offspring population at t^{th} generation (Q_t). These steps are repeated until the required offspring population size N is generated.

Step 2 - Selection: The two solutions are selected from the solution pool P_t (parent population at generation t). There are three possible cases: (i) both solutions are feasible, (ii) one is feasible and the other is infeasible, and (iii) both are infeasible. In case (ii), the feasible solution is selected, and in (iii), the solution with a smaller constraint violation is selected. In case (i), where both solutions are feasible, we can have two cases: (a) both solutions are non-dominated with respect to one another (i.e., both belong to the same $\mathscr{F}_i$) and (b) one dominates the other with respect to one another. In case (b), we select the non-dominated solution, and in case (a), we select the solution using the crowded-comparison operator described in Deb et al. (2002).

Step 3 - Crossover: Two individual solutions are selected using the selection operator, and the crossover sites for each of them are randomly selected. The offspring (a new solution) is created using the genes at the crossover sites.

Step 4 - Mutation: Random genes are inserted into the offspring (a solution obtained in the crossover step) with probability α.

Step 5 - Combining parent (P_t) and offspring (Q_t) population: The offspring solution set (Q_t) generated by repeatedly performing *Step 2* to *4* for N times, is combined with the parent solution set (P_t), the resultant set is R_t which has $2N$ solutions.

Step 6 - Assigning a fitness (or rank): Fitness is assigned to the solutions in set R_t, in the same way as explained in *Step 1*.

Step 7 - Select N best individuals: The N best solutions are chosen. If the size of $\mathscr{F}_1$ is less than N, then all the solutions belonging to $\mathscr{F}_1$ are chosen. This is subsequently done for all the fronts until the population of size N is achieved. It may happen that not all solutions from $\mathscr{F}_i$ can be accommodated into the new population of size N; then, in that case, solutions that need to be included in the population are selected using the crowding distance sorting.

Step 2 to *7* are repeated until the stopping criterion is achieved. The final solution set is then returned.

4.5 Experimental setup

We collaborated with BMW Manufacturing Company (BMW MC) located in Spartanburg, South Carolina for this study. When we started this project at BMW MC, they faced higher detention days for the sea-cans than for the trailers. Therefore, our modeling focus was initially only on sea-cans. However, the inclusion of the warehouse utilization constraint means that an accurate model must incorporate trailers as well since the warehouse is filled with parts received from both sea-cans and trailers. To showcase the impact of including trailers in the model, we decided to run our model twice for each solution strategy, once considering only sea-cans data and once considering both sea-cans and trailers (this latter model is referred to as the *joint model*). Furthermore, we investigate the benefits of including the maximum number of trailers and sea-cans unload capacity constraint. The experiment was run once considering the maximum number of trailers and sea-cans unloading capacity constraint (referred to as '*Capacitated model*') and once by relaxing this constraint (referred to as '*Uncapacitated model*').

The NSGA-II method tries to generate the Pareto optimal solutions as shown in the Figure 4.4 to 4.6. These Pareto fronts were generated for the problem considering a small dataset consisting of eight different parts and thirteen containers (i.e., sea-containers and trailers). We can observe from these three figures that the population size and the number of generations play a vital role in generating the Pareto front. Having a higher population size and the number of generations is desirable. However, it increases the computational time required to generate the final solutions.

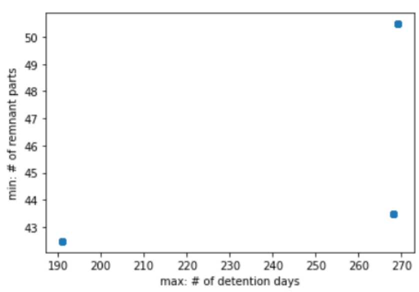


Figure 4.4: Population size = 30 and max number of generation = 30

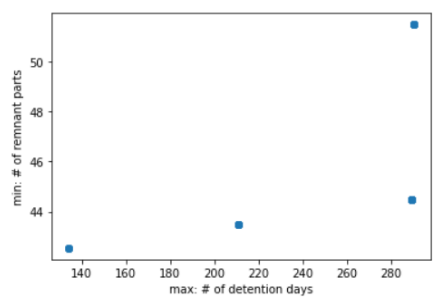


Figure 4.5: Population size = 60 and max number of generation = 30

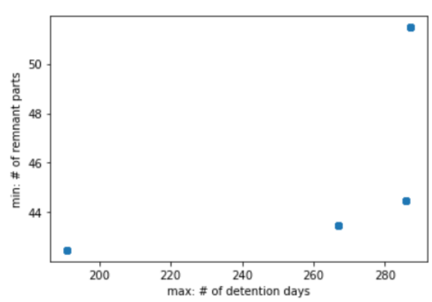


Figure 4.6: Population size = 50 and max number of generation = 250

The optimality of the solutions generated using the NSGA-II cannot be guaranteed as the NSGA-II is a metaheuristic approach. In contrast, the solutions obtained using the ε-constraint method are all Pareto optimal solutions. Therefore, we compared both the methods based on the computational time requirement and the quality of the solutions generated. We ran both methods using a test data set for the comparative study with a one-hour time limit. The test dataset consisted of more than four thousand different parts and more than five hundred containers. The solutions obtained using the ε-constraint method are shown in Figure 4.7. The ε-constraint method took approximately thirty-one minutes to generate these solutions. We ran the NSGA-II method for the initial population of size thirty and iterated it for thirty generations. However, using the NSGA-II method, we could not complete a single iteration (generation) within one hour. Consequently, we decided not to use the NSGA-II method because of the high run-time and the fact that the solutions generated are not guaranteed to be Pareto optimal.

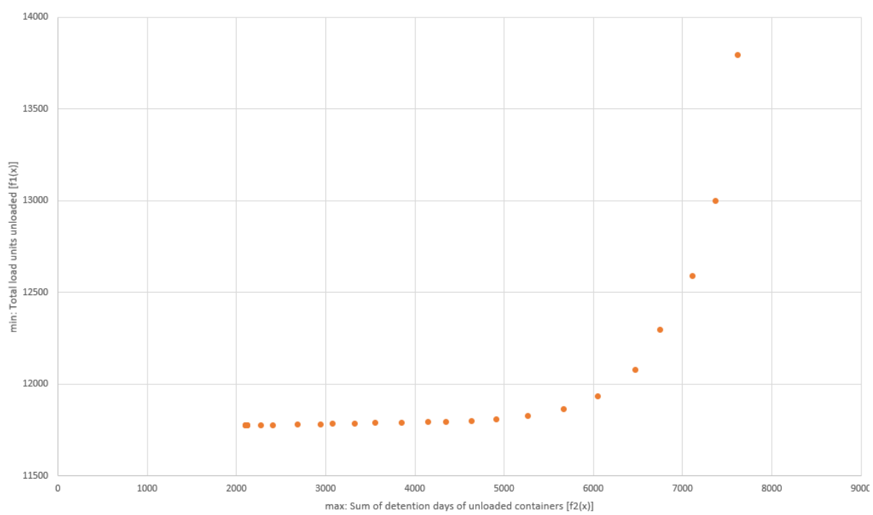


Figure 4.7: Pareto front generated using the ε-constraint method

In general, using the ε-constraint method requires further experimentation than the goal programming method because it involves solving the model with multiple values of ε to generate a Pareto front. A decision-maker then selects one Pareto optimal solution for implementation from the set of the generated Pareto optimal solutions. While running the model with the ε-constraint method, we decided to run it using, (i) minimum feasible ε value and (ii) maximum feasible ε value. The complete list of experiments we ran is presented in Table 4.2

Table 4.2: Experimental models descriptions

	ε-constraint method	Goal programming method
Capacitated	Minimum feasible ε value • Sea-cans only • Joint Maximum feasible ε value • Sea-cans only • Joint	• Sea-cans only • Joint
Uncapacitated	Minimum feasible ε value • Sea-cans only • Joint Maximum feasible ε value • Sea-can only • Joint	• Sea-cans only • Joint

For preliminary analysis, all models were solved for five successive days. The primary purpose of the preliminary analysis was to decide on the most appropriate model for final implementation in BMW MC. We analyzed each model on the following performance measures: (a) The number of sea-cans and trailers needed to be unloaded and predicted to be unloaded, and (b) the set of sea-cans and trailers needed to be unloaded and predicted to be unloaded. Furthermore, using the obtained results, the prediction accuracy of each model is computed. The output of all models is described in the *Results* section.

4.6 Data requirements

Before discussing the results of our experimental study, we provide details on the data required to run these models.

- We used the demand data for each part required to support the production process. Using this data, we determine the minimum stock requirements of a given part in the warehouse.
- We used the data on the quantity of each part available on individual containers/sea-cans available in

the yard for unloading. This data-set includes the containers to be delivered on the first day as well as the containers expected to be delivered on the second and third day of the three-day rolling horizon.

- The packaging data for all the parts stored in the warehouse is also used. This data-set helps determine the load unit quantity that would be unloaded from the container and the space occupied inside the warehouse by the unloaded material from the container.
- The detention days data for each container.

These data sets were readily available via the BMW MC's Enterprise Resource Planning (ERP) system.

4.7 Results

The results for all twelve models are presented below in Table 4.3 and Figures 4.8 - 4.19 for five consecutive production days. Table 4.3 provides the prediction accuracy values for different models. To compute the prediction accuracy, we run the model on *Day* i and obtain the list of containers predicted to be unloaded on next two days (i.e. *Day* $(i+1)$ and *Day* $(i+2)$). Next, we run the model on *Day* $(i+1)$ and obtain the list of containers to be unloaded on *Day* $(i+1)$ and containers predicted to be unloaded on *Day* $(i+2)$. We then compute the total number of predictions made on *Day* i, D, and the number of accurate predictions made, d, using the two list obtained by solving the model on *Day* i and *Day* $(i+1)$. The prediction accuracy is then given by d/D.

Table 4.3: Overall and one day ahead prediction accuracy of all models

Model	Overall	One day ahead
Uncap. ε-con (max. ε value) (sea-cans)	44.94%	81.67%
Cap. ε-con (max. ε value) (sea-cans)	45.29%	69.23%
Uncap. ε-con method (max. ε value) (joint)	41.89%	69.04%
Cap. ε-con method (max. ε value) (joint)	51.66%	56.95%
Uncap. ε-con method (min. ε value) (sea-cans)	73.91%	76.74%
Cap. ε-con method (min. ε value) (sea-cans)	72.46%	75.00%
Uncap. ε-con method (min. ε value) (joint)	81.07%	79.10%
Cap. ε-con method (min. ε value) (joint)	90.66%	88.01%
Uncap. Goal Programming (sea-cans)	68.38%	80.33%
Cap. Goal Programming (sea-cans)	68.38%	80.33%
Uncap. Goal Programming (joint)	72.85%	78.74%
Cap. Goal Programming (joint)	74.97%	79.58%

Figures 4.8 - 4.19, provides the visualization on number of actual and predicted unloading containers for each of the experimental models for the initial study. Each plot shows the number of sea-cans or sea-cans and trailers unloaded on a given day, along with its prediction a day before as well as two days before except for Day 1, Day 2, Day 6, and Day 7. Since the models were run only for five consecutive days, for Day 1, the plot shows only the number of containers actually unloaded, whereas, for Day 7, we only have unload prediction two days before. Similarly, for Day 2, we have unload prediction one day before and actual unload; for Day 6, we have unload prediction one day before and unload prediction two days before. In each figure, we have up to three bars associated with each day and container type (i.e. sea-cans or trailers). Light gray bar depicts the number of containers (sea-cans or trailers) actually unloaded on *Day i* obtained by solving the model on *Day i*, medium gray bar depicts the number of containers predicted to be unloaded on *Day* (i) obtained by solving the model on *Day* $(i-1)$, and dark gray for the number of containers predicted to be unloaded on *Day* (i) obtained by solving the model on *Day* $(i-2)$. Thus, plots where the height of all three bars are close to each other, indicate a higher prediction accuracy.

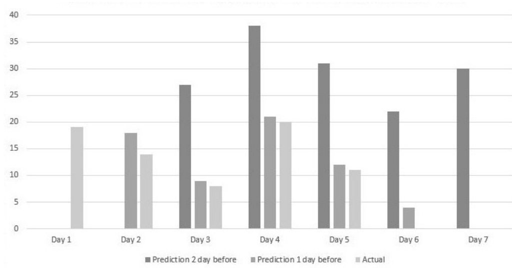


Figure 4.8: Uncapacitated ε-constraint method for sea-cans using maximum ε value

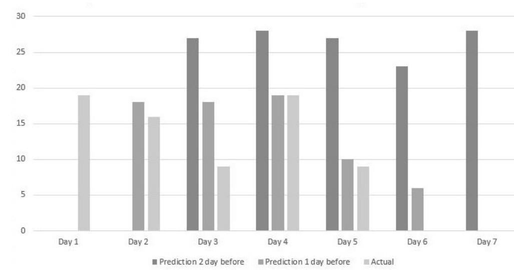


Figure 4.9: Capacitated ε-constraint method for sea-cans using maximum ε value

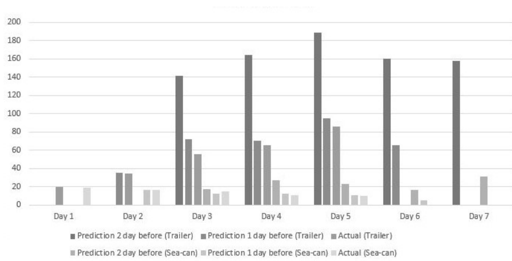


Figure 4.10: Uncapacitated ε-constraint method for sea-cans and trailers using maximum ε value

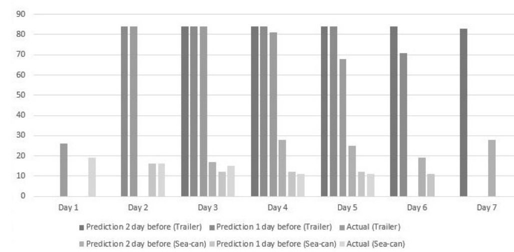


Figure 4.11: Capacitated ε-constraint method for sea-cans and trailers using maximum ε value

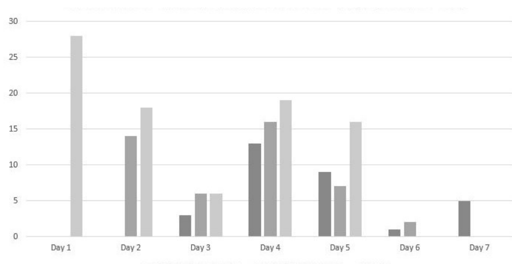


Figure 4.12: Uncapacitated ε-constraint method for sea-cans using minimum ε value

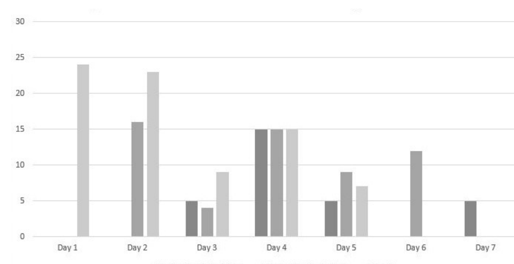


Figure 4.13: Capacitated ε-constraint method for sea-cans using minimum ε value

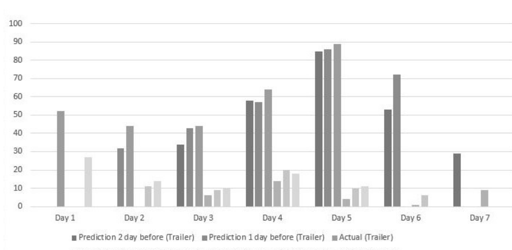


Figure 4.14: Uncapacitated ε-constraint method for sea-cans and trailers using minimum ε value

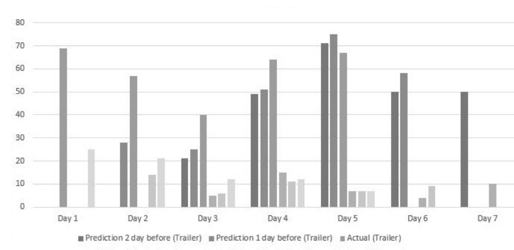


Figure 4.15: Capacitated ε-constraint method for sea-cans and trailers using minimum ε value

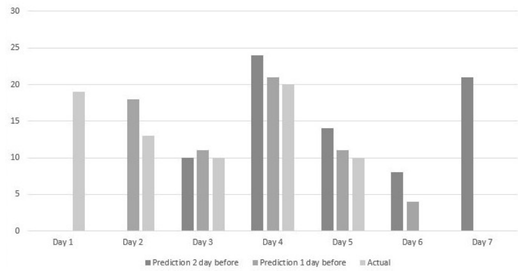


Figure 4.16: Uncapacitated Goal Programming for sea-cans

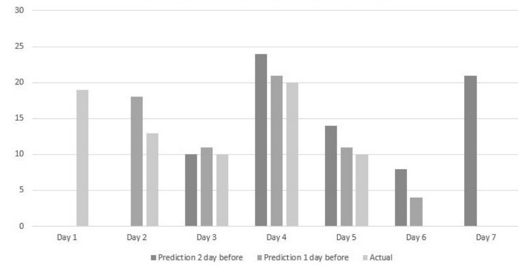


Figure 4.17: Capacitated Goal Programming for sea-cans

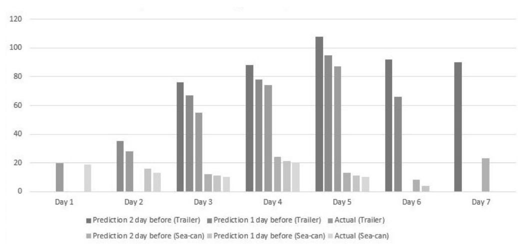


Figure 4.18: Uncapacitated Goal Programming for sea-cans and trailers

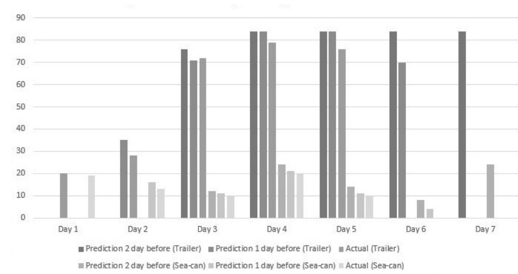


Figure 4.19: Capacitated Goal Programming for sea-cans and trailers

From the above results, we can observe that the performance for the goal-programming models is in between the ε-constraint models that use the maximum and minimum ε values. The goal programming model provides a single optimal solution compared to the multiple Pareto optimal solutions generated by the ε-constraint method. In addition, using the minimum ε value puts more weight on minimizing the remnant parts objective, whereas using the maximum ε value puts more weight on the maximization of the detention objective. Choosing a single ε value for the implementation that would provide a fair trade-off between both objectives requires additional complicated analysis. The decision-maker is not interested in obtaining multiple Pareto optimal solutions generated using the ε-constraint method and choosing one on a daily basis. Additionally, satisfying the capacity constraint is important for the decision-maker at BMW MC, because it helps them operate successfully using the available workforce and other warehouse resources such as forklifts. Based on these observations and discussions, we decided to do the pilot study using the capacitated goal programming method for sea-containers and trailers.

4.8 Pilot study

After selecting a model, we had to validate its performance over a longer period and fix any issues that might cause a problem after implementing it into the production system. For this, we did a pilot study for a period of 21 consecutive production days using the selected goal programming model. During the selected period for the pilot study, the production was ramping up to the normal capacity after returning to normalcy from COVID-19 related disruptions. Therefore, we got an opportunity to review the proposed model's performance during adversities. Figure 4.20 represents the decision tree used to decide on the unload capacity constraint to be incorporated while solving the optimization model. In the Figure 4.20, T_1 and S_1 denote the decision maker's ideal unloading capacity for trailers and sea-cans, respectively. In case the optimization model is infeasible when considering T_1 and S_1 as unloading capacity values, then the decision maker's second preferred unloading capacity values T_2 and S_2 are used. If the model remains infeasible when T_2 and S_2 are used as unloading capacity, then the optimization model is solved by relaxing the unloading capacity constraint.

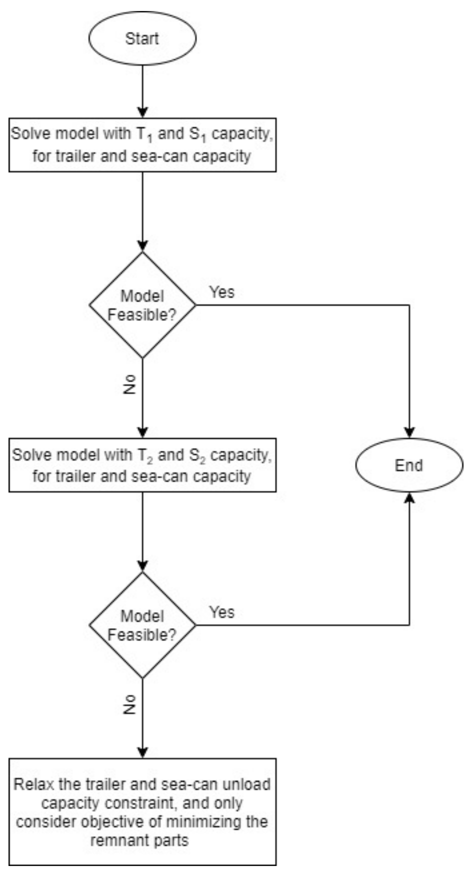


Figure 4.20: Decision flow of selected model

The output values for number of trailers and sea-cans unloaded on each day is shown in the Figure 4.21 to Figure 4.24. The model remains feasible for every instance because we systematically relax the constraints that make it infeasible, as given in Figure 4.20. From the 20 instances for which the model was run, we observed, (i) for the three instances, the model with T_2 and S_2 capacity was run, and (ii) only once the model with relaxed unload capacity constraint was run.

In the pilot study, the proposed goal programming model has an overall prediction accuracy of 74.62%. Moreover, the accuracy of one-day ahead predictions is comparable at 73.87%. The overall prediction accuracy for the sea-cans is 57.11%, and for the trailers, it is 77.71%.

We also investigated the daily total load units (LU) unloaded and the daily detention days achieved over the validation study period. The plot for this is provided in Figure 4.25. In this plot, we can observe that the daily total load units (LU) unloaded are stable overall except for Day 14 to Day 17 because the plant was returning to normal operations after a three day holiday shutdown. Also, the daily detention days achieved remain stable over time as observed from Figure 4.25.

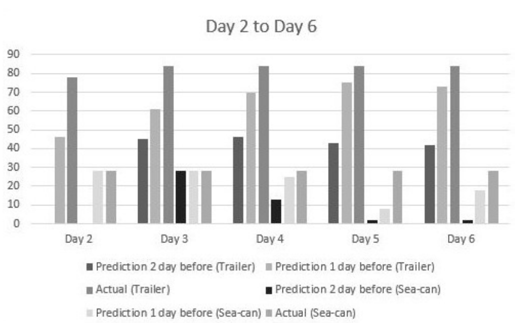


Figure 4.21: Pilot study Day 2 to Day 6

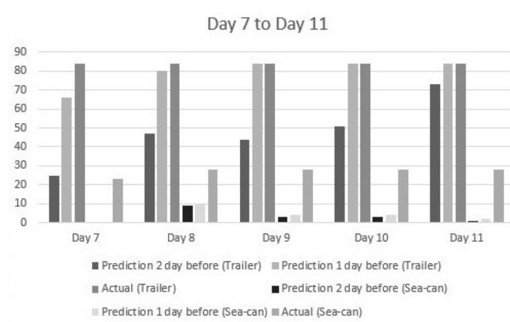


Figure 4.22: Pilot study Day 7 to Day 11

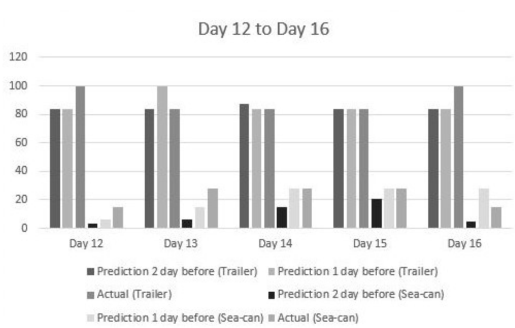


Figure 4.23: Pilot study Day 12 to Day 16

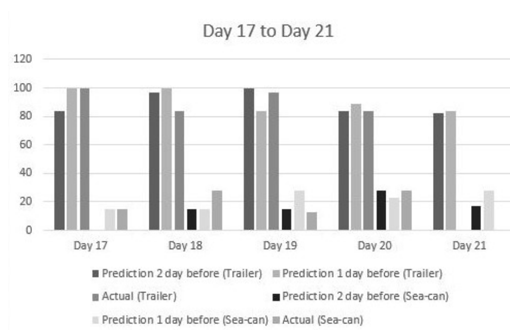
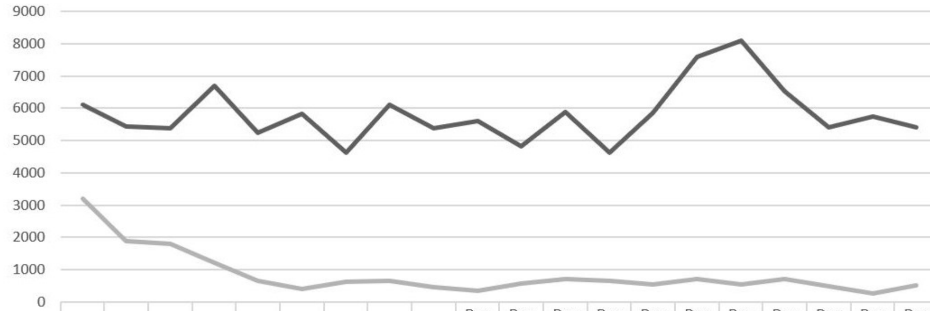


Figure 4.24: Pilot study Day 17 to Day 21



	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15	Day 16	Day 17	Day 18	Day 19	Day 20
Total load units unloaded	6107	5444	5382	6707	5244	5838	4622	6107	5377	5605	4824	5877	4628	5858	7590	8088	6520	5398	5729	5411
Total detention days achieved	3202	1893	1805	1216	663	393	615	648	468	351	577	700	641	540	698	533	697	490	257	521

Figure 4.25: Daily LU unloaded and total detention days achieved

4.9 Observations and conclusions

The results show that the highest value of overall prediction accuracy is obtained from the model using the minimum ε value for the capacitated model with the joint case. When the minimum ε value is chosen, there are no restrictions on the average detention days that must be attained by the set of sea-cans and trailers selected. In this case, the problem becomes a single objective optimization model to minimize the number of remnant parts in the warehouse. Further, we observe the difference in the prediction accuracy for the uncapacitated models versus the capacitated models for all methods. This difference is due to the change in the selection pattern of the sea-cans and trailers caused by the additional unload capacity constraint.

The solution corresponding to the minimum ε value places more importance on minimizing the remnant parts objective. Whereas, the solution corresponding to the maximum ε value places more importance on maximizing the detention penalty objective. The goal programming model generates a single solution with an appropriate trade-off between both objectives. In our problem, the decision-maker places equal importance on both the objective and is interested in obtaining a single solution. Therefore, we selected the goal programming model with unloading capacity constraints for the pilot study.

Sometimes, it might not be possible to satisfy the unloading capacity constraint for sea-cans and trailers. Consequently, in the pilot study, we developed a logic by systematically updating and relaxing the unloading constraints such that always a feasible solution is produced. In our pilot study, we analyzed the performance of the proposed goal programming model for 20 consecutive production days. In the pilot study, we observed that the total load units unloaded remain stable on the regular operating days. Whereas the daily detention days achieved remained stable over time.

In December 2017, BMW MC faced serious operational issues due to the overutilization of the warehouse, which led to the shutdown of the entire assembly line. These issues were caused by the failure of the current container selection method used by BMW MC. The current method used by BMW MC uses a heuristic to calculate the available days of supply in the warehouse and only consider the present moment's demand. Warehouse utilization is not considered while deciding on unloading containers. Further, the decision-maker often intervenes by overruling the decision taken by the current system.

The proposed model overcomes these issues experienced with the current container selection process at BMW MC. Further, it provides the decision-maker with future unloading information, which is vital in planning the warehouse operations. The model also ensures that the warehouse utilization is maintained below the desired threshold on all three planning days considered in the rolling horizon.

Chapter 5

Team Formation and Job Assignment to Improve Warehouse Operations

5.1 Introduction

In a warehouse, inventory gets depleted as the parts are consumed to support the manufacturing processes. Therefore, the inventory needs to be replenished regularly using incoming shipments of parts. We consider a manufacturing plant that receives parts in sea-containers and trailers, and stores them on site until a part replenishment is needed. Multiple sea-containers and trailers, which from here on we refer to as containers, are often received daily by the manufacturing plant, and each container may hold a different mixture of parts with potentially different packaging. Thus, the unloading of parts from these containers and their storage in the warehouse may require unpacking of pallets and/or repacking of materials on a daily basis before they can be scanned, transported, and stored. As it is often the case that workers have limited physical space (i.e., one or more staging areas) for unloading, unpacking, and scanning materials, completing these tasks often requires coordination between multiple workers to avoid backlogs and disruption of production. Such coordination can be achieved by having workers work in teams to process contents of each container.

Optimization of the put-away operations is vital for timely processing and storage of materials delivered to the warehouse. Therefore, our goal in this research is to minimize the total time required to unload, scan, breakdown, and store all pallets from containers that must be unloaded by determining the optimal number of teams to be formed and the optimal worker to task assignment in these teams. To address these operational questions, we build a simulation-optimization model and analyze it using the daily container and processing time data obtained from our industrial partner, BMW Manufacturing Company.

In the next section, we review the existing literature on optimizing warehouse operations, resource allocation in healthcare and manufacturing, and studies that consider worker differences in workforce planning. In section 5.3, we provide the problem description, details of the simulation model we built, and formulated optimization model. We discuss the experimental setup in section 5.4. In section 5.5, we present the numerical results and discussion on the observations made. Finally concluding remarks are provided in section 5.6.

5.2 Literature review

There are several previous studies on the optimization of warehouse operations. However, very few of these studies have focused on receiving and shipping processes that are part of the warehouse operations (Gu et al., 2007). Further, most of the studies related to receiving and shipping processes are concerned with truck-to-door assignment problems in cross-dock facilities (Gu et al., 2007). An exhaustive review of inventory replenishment problems that were addressed using simulation-based optimization methods is provided by Jalali and Nieuwenhuyse (2015). In addition, Negahban and Smith (2014) performed a comprehensive review of studies on manufacturing design and operations-related problems that were tackled via discrete event simulation. According to their review, manufacturing operations planning and scheduling is a popular area for the use of discrete event simulation as an analysis tool. In this section, we review the simulation studies that have focused on optimizing worker and resource allocation to improve healthcare, manufacturing, and warehouse systems.

Resource allocation problems in healthcare systems have been widely studied using simulation-based optimization methods. Weng et al. (2011) used simulation optimization to study a physician and nurse allocation problem in the emergency department of a hospital to minimize the NEDOCS value, which is an indicator of crowdedness. Optimal staffing levels that balance the service level (i.e., quality of care) and nurse utilization in each shift was studied by Sarno and Nenni (2016). Lucidi et al. (2016) studied the optimal allocation of a hospital's obstetrics ward resources such as stretchers, gynecologists, nurses, beds, and operating rooms. Their objective was to maximize the profits of the hospital and minimize the cesarean section birth rates.

A study on optimal resource allocation for a production logistics system was done by Li et al.

(2020) using discrete event simulation. Their objective was to maximize the throughput of the system while minimizing the economic input, which is achieved by determining the optimal number of automatic guided vehicles (AGVs), speed and load capacity of AGVs, and buffer capacity. Ekren et al. (2012) studied the problem of reducing the cycle time in the receiving area of a warehouse by determining the optimal number of workers at each workstation. The truck's arrival to the warehouse is considered deterministic. At the same time, each truck's contents are probabilistically known. A simulation-based optimization model is proposed to decide the optimal staffing level at each station. In our study, we also minimize the total completion time for the put-away operations. However, we achieve this objective by determining the optimal number of teams (parallel workstations) to be formed, the number of workers and worker to task assignment in each team, buffer space allocation, and the number of forklift operators needed. Ganbold et al. (2020) studied an optimal workforce allocation problem to improve the warehouse service level using a simulation-based optimization model. They considered a warehouse with inbound and outbound areas, each with various workstations. In their model, there are multiple employees with different skills, and the goal is to optimally allocate these employees to workstations while considering the warehouse operational constraints. However, the constraints related to warehouse storage capacity, working areas, and buffer zones between activities are not considered in this problem. In contrast, we determine the optimal number of parallel workstations (teams of workers) that need to be created, the number of workers at each workstation, and the buffer space for each workstation.

Van Oyen et al. (2001) studied the benefits of a flexible workforce in a collaborative and non-collaborative environment in a serial production system. In a collaborative environment, multiple workers can simultaneously work on a job at a time. In contrast, in the non-collaborative environment, only one worker can work on a job at a time. Andradóttir and Ayhan (2005) studied a flexible server assignment problem to maximize the throughput of a serial production system. In this problem, the authors considered a collaborative work environment and studied the server assignment policies for the two-station tandem system. Işık et al. (2016) exhaustively studied the problem of flexible server assignments in a non-collaborative work environment for a tandem queueing system with an equal number of stations and servers. Similar to our study, the authors in Andradóttir and Ayhan (2005) also studied the assignment policies that involve grouping available servers into two or three teams and assigning them to two stations. However, in contrast to the study done by Andradóttir and Ayhan (2005), in our problem, we determine the number of teams (parallel queues) that need to be formed and the server assignments and resource allocations to the different stations for each queue formed. Further, for the problem we consider in this chapter, the work environment at some stations in a parallel queue is collaborative, and at others, it is non-collaborative.

Studies that consider the worker differences while solving the workforce planning problem have been studied widely. Askin and Huang (2001) studied a team formation and training problem for cellular manufacturing. The authors used the Kolbe Conative Index (KCI) in the problem formulation to estimate an individual worker's conative aptitude and team synergy. A multiobjective problem was formulated and solved using a weighted average method to minimize (1.) the training cost, (2.) the misfit between worker's conative traits and job requirements, and (3.) deviations from desired team synergy. Wirojanagud et al. (2007) studied a workforce planning problem by explicitly considering worker differences using the General Cognitive Ability (GCA) metric in the problem formulation. A Mixed Integer Program (MIP) was formulated and solved using a solution space partition approach (decomposition approach) to minimize the total workforce-related (i.e., hiring, training, salary, and firing) and missed production costs over multiple periods. In this study, the authors assumed that an individual worker's productivity depends on the GCA level. The computational time required to solve the problem studied by the Wirojanagud et al. (2007) could be high even when using the decomposition approach proposed by the authors. Therefore, Fowler et al. (2008) developed Linear Programming based heuristics to reduce the computational time required to solve the problem studied by the Wirojanagud et al. (2007). In this study, we do not consider the differences between the individual workers by assuming all of them to be identical.

5.3 Problem description and simulation model

Containers are delivered to the manufacturing plant daily to support the manufacturing process. Each container holds multiple pallets containing a mixture of different parts with potentially different packaging. The set of containers that need to be unloaded is known at the beginning of the day. Additionally, each container is assigned a priority level for unloading such that containers that have the parts with the least supply in the warehouse are assigned the highest priority. The manufacturer follows a defined procedure to unload these containers in order to replenish warehouse stock. A container is docked, pallets are unloaded and processed in a primary staging area, and then they are transported either to the warehouse for storage or to a secondary staging area if they need further processing. In the primary staging area, two types of workers, known as the *Scanners* and the *Breakdowners*, process the unloaded pallets. A scanner must scan all unloaded pallets. Some pallets may also need to be broken down into individual boxes by a breakdowner. The entire process is illustrated in Figure 5.1.

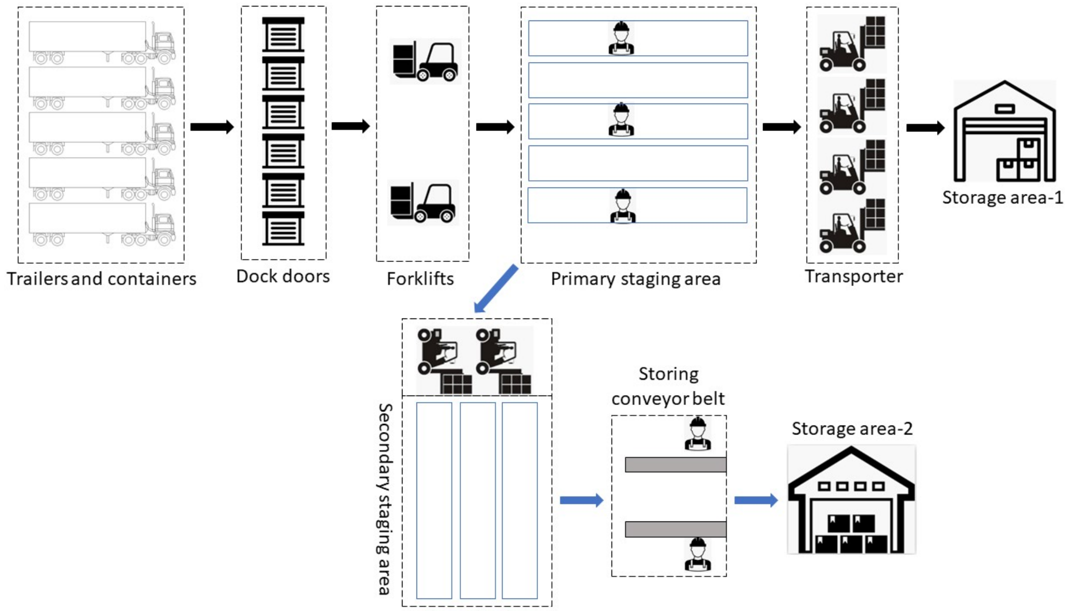


Figure 5.1: Process flow chart

In this chapter, we only focus on the part of the receiving process until the parts are ready to be transported out of the primary staging area. The equivalent queuing representation for this process is shown in Figure 5.2. Process 1 has a finite number of containers whose contents need to be unloaded and processed. Different containers that need to be unloaded are represented using different colored trapezoids in Figure 5.2. The number of parallel queues in Process 1 is equivalent to the number of teams. Queues at Process 2 are capacitated, and once again, the total number of parallel queues is equivalent to the number of teams. The capacity of each queue in Process 2 is equal to the buffer size (i.e., number of staging areas) allocated to each team. Not all pallets unloaded from a container are identical. In Figure 2, various shapes are used to differentiate different types of pallets waiting in a queue at Station 2. Additionally, pallets unloaded from the same container are depicted using the same color. Two different teams are represented in Figure 5.2 with the colors orange and purple. Figure 5.2 illustrates the entire procedure with two teams; however, the number of teams that need to be formed is a decision variable in our problem.

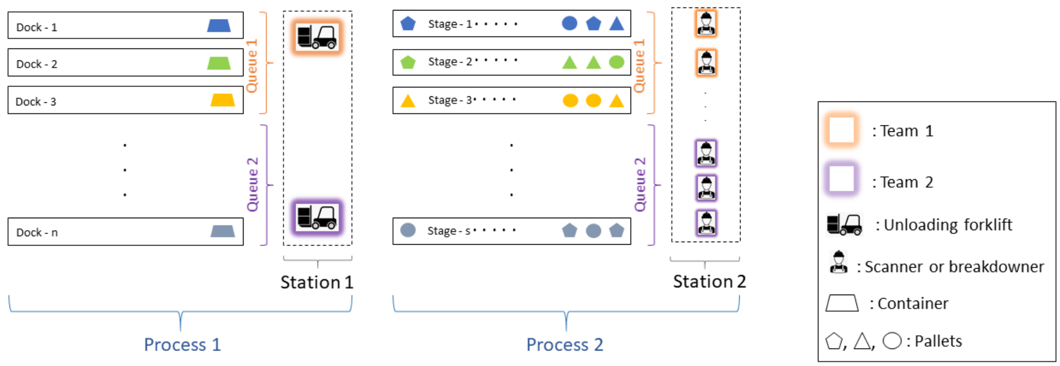


Figure 5.2: Queue representation of the process

One or more unloading forklifts work at Station 1 to unload the pallets from the containers. At *Station 2*, scanners and breakdowners work in teams to process (scan and breakdown) the unloaded pallets. The order of processing for each container is determined by its priority level and known in advance. Only one forklift can work on a container to unload all its contents. The unloaded pallets are then placed into the staging area. Contents of a container can be placed in any of the available staging areas, however, a staging area cannot simultaneously hold pallets from different containers. Once a container is completely unloaded, the scanner starts scanning the unloaded pallets. After all pallets of a container are scanned, the breakdowners can break down the pallets if required. The scanner starts scanning the pallets from a new container while the breakdowners work on pallets from the previous container. The scanners and breakdowners work in a team. In each team, there is one scanner and one or more breakdowners. Collaboration between teams is not permitted. Furthermore, once the teams are formed at the beginning of the day, they cannot be changed until all containers are processed. At the beginning of each day, every team is assigned a set of staging areas to be used throughout the day. When the scanning and breakdown operations are completed for a container, the transporter forklifts transfer the processed pallets to the warehouse storage spaces and clear the staging area. The transporter forklifts can work for any team. Similar to the unloading forklifts working at Station 1, only one transporter forklift can transport the materials belonging to a container.

On a given day, there are S staging areas available, as well as T transporter forklifts, and N workers who can serve as a scanner, breakdowner, or an unloading forklift's driver. Teams must be formed optimally using the available resources (i.e., staging areas and workforce) to minimize the time required to process and store the materials from the containers that must be unloaded for the day. We develop a simulation-based optimization model to solve this problem. A detailed discussion of the simulation model is provided below.

The simulation model is built in Arena, and OptQuest is used to build the optimization model. Figure 5.3 shows a representation of the simulation model we developed. The process starts with the arrival of all containers (i.e., jobs). The containers enter the system in priority order. A container is held until a forklift and a buffer space (i.e., staging area) are available to unload it. This process is executed using the Hold module labeled as *Hold containers* in Figure 5.3. Once a forklift and buffer space is available, the Process module labeled as *Sieze forklift and buffer space* seizes one unit of both resources for a container. The Process module labeled as *Release forklift* releases the forklift after an Erlang distributed amount of time, once all pallets from the container are unloaded. The service times per pallet for a forklift, scanner, and breakdowner is assumed to be exponentially distributed, therefore, the service time for a container containing *n* pallets is Erlang distributed. Our industrial partner had conducted the time study for each of these servers (i.e., forklift, scanner, and breakdowner) and provided us with the mean service time for each process.

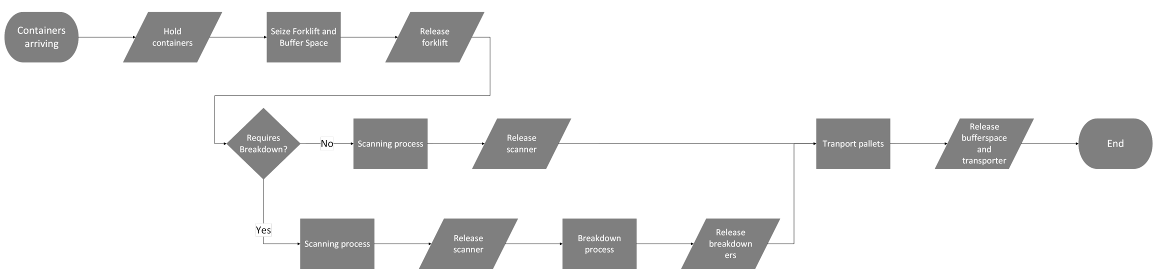


Figure 5.3: Representation of the simulation model

As mentioned previously, each team is allocated buffer spaces within which they are allowed to work. Therefore, if a container is unloaded into the buffer space of a team, then only members of that team can process it. Using the Assign module in Arena, we keep track of the team assignments. The job is then routed to the specific team that needs to process it. Scanning is required for all pallets in a container. However, there are two possibilities for the breakdown: (i) breakdown is not required for any of the pallets in a container, and (ii) breakdown is required for some or all pallets in a container. We use the Decide module labeled as *Requires breakdown* to route the job appropriately. In case (i), only scanning is required, and therefore only the scanner resource is seized. Process times for scanning follow an Erlang distribution. In case (ii), first, the scanner is seized to scan the pallets and released after and Erlang distributed amount of time. Then, the breakdowners are seized to breakdown the pallets, and they are released after an Erlang distributed amount of time. After scanning and breakdown of all pallets belonging to a container is completed, the transporter forklift is seized using the Process module labeled as *Transport pallets* to clear all materials from the staging area. Process times for transportation also follow an Erlang distribution. Finally, the buffer space is released upon completion of the transporter forklift process. A summary of the process time distributions per container used in the model is provided in Table 5.1.

Table 5.1: Summary of the process time distributions

Process	Distribution	Shape parameter
Unload	Erlang distribution	Total pallets in a container
Scanning	Erlang distribution	Total pallets in a container
Breakdown	Erlang distribution	Total breakdown pallets in a container
Transporting to warehouse shelves	Erlang distribution	Total non-breakdown pallets in a container
Transporting to secondary staging	Erlang distribution	Total breakdown pallets in a container

Note: Rate parameter of each Erlang distribution was estimated by the industry partner using propriety data.

Using the input parameters defined in Table 5.2 and the decision variables defined in Table 5.3, we formulate the optimization model given by equations (5.1) – (5.13).

Table 5.2: Input parameters notation and description

Notation	Description
t_{min}	Minimum number of teams to be formed
t_{max}	Maximum number of teams that can be formed
S	Total staging areas
N	Total number of individuals available

Table 5.3: Decision variables notation and description

Notation	Description
s_i	Binary variable, equals to 1 if *Team i* is formed. $(where\ i = 1,\ 2, \dots\ t_{max})$
b_i	Number of breakdowners in *Team i*. $(where\ i = 1,\ 2, \dots\ t_{max})$
x_i	Number of staging areas allocated to *Team i*. $(where\ i = 1,\ 2, \dots\ t_{max})$
f	Number of unload forklifts

$$\text{minimize:} \quad \textit{Expected total time required to process and put-away all the containers} \tag{5.1}$$

Subject to:

$$\sum_{i=1}^{t_{max}} s_i \geq t_{min} \tag{5.2}$$

$$\sum_{i=1}^{t_{max}} s_i \leq t_{max} \tag{5.3}$$

$$b_i - s_i \geq 0 \quad \ldots i \in \{1,\, 2, \ldots,\, t_{max}\} \tag{5.4}$$

$$x_i - s_i \geq 0 \quad \ldots i \in \{1,\, 2, \ldots,\, t_{max}\} \tag{5.5}$$

$$s_i * b_i - b_i = 0 \quad \ldots i \in \{1,\, 2, \ldots,\, t_{max}\} \tag{5.6}$$

$$s_i * x_i - x_i = 0 \quad \ldots i \in \{1,\, 2, \ldots,\, t_{max}\} \tag{5.7}$$

$$f + \sum_{i=1}^{t_{max}} s_i + \sum_{i=1}^{t_{max}} b_i = N \tag{5.8}$$

$$\sum_{i=1}^{t_{max}} x_i = S \tag{5.9}$$

$$s_i \in \{0,\, 1\} \quad \ldots i \in \{1,\, 2, \ldots,\, t_{max}\} \tag{5.10}$$

$$b_i \in \mathbb{Z}_{\geq 0} \quad \ldots i \in \{1,\, 2, \ldots,\, t_{max}\} \tag{5.11}$$

$$x_i \in \mathbb{Z}_{\geq 0} \quad \ldots i \in \{1,\, 2, \ldots,\, t_{max}\} \tag{5.12}$$

$$f \in \mathbb{Z}_{>} \tag{5.13}$$

Constraints defined by equations (5.2) and (5.3) provide a lower and upper bound on the number of teams that need to be formed. When *Team i* is formed, it is assigned a scanner using the decision variable s_i. If *Team i* is formed, then it gets assigned a minimum of one breakdowner and one buffer space; constraints (5.4) and (5.5) ensure this. Further, if *Team i* is not formed, then no breakdowner and buffer space are assigned to *Team i* due to constraints (5.6) and (5.7). Constraint (5.8) ensures that all the available individuals are assigned, and constraint (5.9) ensures that all the buffer spaces are allocated to teams.

5.4 Experimental setup

We analyzed the data provided by our industrial partner for the number of containers unloaded on a given day in the warehouse. We observed that around eight percent of the time, the number of containers unloaded was less than 75. Most of the time, around 100 containers were unloaded daily, with the maximum being 115 containers unloaded on a single day. Therefore, we chose to run our experiments using 75, 100, and 125 as the number of containers to be unloaded. Further, by discussing with the process associates at the BMW Manufacturing Company, we decided to vary the percentage of containers requiring breakdown as 25%, 50%, and 75%. Using this information, we construct nine scenarios for our simulation experiments. The number of pallets in each container is generated using the discrete uniform distribution, U$\{15, 25\}$, as estimated by our industrial partner. The percentage of pallets to be broken down in each container that requires breakdown is estimated to follow the continuous uniform distribution $U[30, 70]$ based on the analysis carried out by our industry partner.

On each day, 14 individuals are available to work as either a forklift operator, scanner, or breakdowner. A minimum of 2 and a maximum of 4 individuals need to be assigned as forklift operators. The remaining individuals need to be assigned to a minimum of 3 and a maximum of 5 teams, each team consisting of one scanner and at least one breakdowner. There are 13 staging areas available that need to be distributed amongst the teams. These details on capacity of resources and process requirements were provided by the industrial partner.

The optimization model was constructed in OptQuest using the parameters mentioned above. The number of solutions to be explored in OptQuest was set as 100. Additionally, the number of replications performed for each simulation run was dynamically determined by OptQuest using the 95% confidence interval built around the mean of the total time required to complete all jobs, with a minimum of 30 and a maximum of 50 replications. The best solution obtained by OptQuest was then evaluated using the simulation model with 100 replications. We test the solutions obtained using OptQuest, against the currently practiced team formation. For this purpose, we simulated the unloading process with the current team formation under each scenario considered. These simulations were also run using 100 replications.

5.5 Results

In this section, we discuss the results obtained using the simulation optimization model. In Table 5.4, the currently practiced team formation is presented. Currently, the same team formation with three teams and two forklifts is used every day, regardless of the number and characteristics of containers to be unloaded. Each team has an equal number of breakdowners and an approximately equal number of buffer spaces allocated. The best solutions (i.e., solutions with a minimum total time for processing all jobs) for nine different scenarios obtained using the OptQuest are presented in Table 5.5.

Table 5.4: Currently practiced team formation

Breakdowner allocation			Buffer space allocation			Scanner allocation			Number of forklifts
T_1	T_2	T_3	T_1	T_2	T_3	T_1	T_2	T_3	
3	3	3	4	4	5	1	1	1	2

Table 5.5: OptQuest solution for best team formation

Breakdown (%)	Number of containers	Breakdowner allocation				Buffer space allocation				Scanner allocation				Number of forklifts
		T_1	T_2	T_3	T_4	T_1	T_2	T_3	T_4	T_1	T_2	T_3	T_4	
25	75	2	1	2	1	8	2	2	1	1	1	1	1	4
25	100	3	1	1	1	8	2	2	1	1	1	1	1	4
25	125	3	1	1	1	8	2	2	1	1	1	1	1	4
50	75	1	1	1	3	2	2	2	7	1	1	1	1	4
50	100	1	1	3	1	2	2	8	1	1	1	1	1	4
50	125	1	1	3	1	2	2	8	1	1	1	1	1	4
75	75	2	1	2	2	2	2	4	5	1	1	1	1	3
75	100	2	1	2	2	2	2	4	5	1	1	1	1	3
75	125	3	1	2	1	4	2	6	1	1	1	1	1	3

Results reported in Table 5.5 show that currently, the forklift operators are a bottleneck. Further, we observe that the optimal team formation is dependent on the breakdown percentage. The optimal team formation remains the same for the scenarios with 25% – 50% breakdown and 100 – 125 containers. A similar observation can be made in the case of 75% breakdown and 75 – 100 containers. Having fewer solutions to implement in different scenarios is better from a practical perspective for our industrial partner. Furthermore, some of the solutions presented in Table 5.5 may be statistically equivalent in terms of performance under a given scenario. To reduce the number of solutions we propose and eliminate statistically equivalent solutions, we have performed 95% t-tests to compare the performance of the solutions in Table 5.5 across scenarios. Based on this analysis, we present the proposed solutions that depend on the breakdown percentage in Table 5.6. The solutions presented in Table 5.5 and Table 5.6 are statistically equivalent as per our analysis.

Table 5.6: Proposed solution for best team formation

Breakdown (%)	Number of containers	Breakdowner allocation				Buffer space allocation				Scanner allocation				Number of forklifts
		T_1	T_2	T_3	T_4	T_1	T_2	T_3	T_4	T_1	T_2	T_3	T_4	
25	75	2	1	2	1	8	2	2	1	1	1	1	1	4
25	100	2	1	2	1	8	2	2	1	1	1	1	1	4
25	125	2	1	2	1	8	2	2	1	1	1	1	1	4
50	75	2	1	2	1	5	2	4	2	1	1	1	1	4
50	100	2	1	2	1	5	2	4	2	1	1	1	1	4
50	125	2	1	2	1	5	2	4	2	1	1	1	1	4
75	75	2	1	2	2	2	2	4	5	1	1	1	1	3
75	100	2	1	2	2	2	2	4	5	1	1	1	1	3
75	125	2	1	2	2	2	2	4	5	1	1	1	1	3

The average time required to complete all jobs under different scenarios using the current team formation (i.e., Table 5.4) and the proposed solution (i.e., Table 5.6) is presented in Table 5.7. These results suggest a minimum 19% and maximum 39% of performance improvement under the various scenarios considered.

Table 5.7: Performance improvement

Breakdown (%)	Number of containers	Currently practised team formation		Proposed solution		Average performance improvement
		Average (min)	Half-width (min)	Average (min)	Half-width (min)	
25	75	545.65	4.85	341.46	4.65	37.42%
25	100	714.40	4.94	434.10	4.26	39.24%
25	125	884.30	4.98	534.52	5.01	39.55%
50	75	554.11	4.37	391.39	5.00	29.37%
50	100	722.14	4.90	504.16	5.27	30.19%
50	125	896.92	4.99	615.53	5.77	31.37%
75	75	558.11	4.17	451.07	4.65	19.18%
75	100	728.99	4.56	582.69	4.77	20.07%
75	125	898.06	4.07	710.43	5.61	20.89%

The details related to the average utilization of each resource under the current team formation and the proposed solution are presented in Table 5.8 and Table 5.9, respectively. We observe that utilizations of breakdowners and buffer spaces increase when the breakdown percentage or number of containers increases. In comparison, the average utilizations of scanners and forklifts increase only with an increase in the number of containers. Additionally, the average utilization per scanner and the average utilization per forklift remain the same for different breakdown percentages as long as the number of containers remain the same. This is because the workloads of the forklifts and scanners are unaffected by the amount of breakdown needed. These results also show that the average utilization per forklift is higher using the presently practiced team formation because of the lower number of forklifts deployed when compared to the proposed solution. Furthermore, the average utilization per scanner is higher under the currently practiced team formation because of the lower number of teams formed when compared to the proposed solution. On the contrary, the average utilizations of breakdowners and buffer spaces under the current team formation is lower than under the proposed solution because improper resource allocation in the current practice results in a longer putaway time required to process all containers. Table 5.10 presents the detailed per resource utilization in each team for all resources under the proposed solution given in Table 5.6.

Table 5.8: Average resource utilization using currently practised team formation

Breakdown (%)	Number of containers	Breakdowner	Buffer space	Scanner	Forklift
25	75	0.056	0.184	0.063	0.425
25	100	0.073	0.245	0.083	0.566
25	125	0.092	0.307	0.104	0.709
50	75	0.111	0.214	0.062	0.426
50	100	0.148	0.284	0.084	0.566
50	125	0.185	0.356	0.104	0.711
75	75	0.164	0.243	0.063	0.425
75	100	0.221	0.326	0.084	0.567
75	125	0.277	0.407	0.104	0.708

Table 5.9: Average resource utilization using proposed solution

Breakdown (%)	Number of containers	Breakdowner	Buffer space	Scanner	Forklift
25	75	0.084	0.226	0.047	0.212
25	100	0.110	0.301	0.062	0.284
25	125	0.139	0.381	0.078	0.354
50	75	0.167	0.271	0.047	0.213
50	100	0.223	0.363	0.062	0.283
50	125	0.279	0.455	0.078	0.354
75	75	0.211	0.305	0.047	0.283
75	100	0.285	0.408	0.063	0.378
75	125	0.351	0.510	0.078	0.475

Table 5.10: Resource utilization for proposed solution

Breakdown (%)	Number of containers	Breakdowner utilization				Buffer space utilization				Scanner utilization				Forklift utiliz-ation
		T_1	T_2	T_3	T_4	T_1	T_2	T_3	T_4	T_1	T_2	T_3	T_4	
25	75	0.16	0.07	0.04	0.03	0.24	0.22	0.21	0.19	0.12	0.03	0.03	0.01	0.21
25	100	0.21	0.09	0.05	0.04	0.31	0.30	0.28	0.26	0.16	0.04	0.04	0.02	0.28
25	125	0.26	0.13	0.06	0.05	0.39	0.38	0.36	0.34	0.20	0.05	0.05	0.02	0.35
50	75	0.21	0.15	0.16	0.13	0.28	0.28	0.26	0.26	0.08	0.03	0.06	0.02	0.21
50	100	0.27	0.19	0.21	0.18	0.37	0.37	0.35	0.34	0.10	0.04	0.08	0.03	0.28
50	125	0.34	0.24	0.26	0.23	0.47	0.46	0.44	0.44	0.13	0.05	0.10	0.04	0.35
75	75	0.14	0.22	0.24	0.26	0.33	0.33	0.31	0.28	0.04	0.03	0.06	0.07	0.28
75	100	0.18	0.29	0.32	0.35	0.43	0.44	0.41	0.39	0.05	0.04	0.08	0.09	0.38
75	125	0.22	0.35	0.39	0.44	0.54	0.54	0.52	0.48	0.06	0.04	0.10	0.11	0.47

Based on the experimental results, we make three important observations. First, the unloading forklifts are a bottleneck in the currently practiced team formation. Therefore, according to our proposed solution, more forklift operators (3–4) must be assigned. Second, the number of workers assigned as breakdowners should increase with the increase in the breakdown percentage. This is because the breakdown process becomes a bottleneck as breakdown percentages increase. The number of breakdowners is increased by reducing the number of forklift operators assigned in our proposed solution. Third, four teams should be formed instead of the three teams formed in the current practice. This is evident from the higher average breakdowner and buffer space utilization achieved using the proposed solutions.

5.6 Conclusions

We study the problem of optimal team formation, buffer allocation, and job assignment to teams of workers to optimize the container unloading and warehouse replenishment process. We solve this problem using a simulation optimization approach. Currently, the industrial partner forms three teams with equal sizes and approximately equal buffer space allocations. The solutions obtained using the simulation optimization model suggest that the team formation should be dependent on the percentage of containers that require additional processing (i.e., breakdown). Further, the unloading forklift operators are the bottleneck in the current process. This bottleneck can be alleviated by increasing the number of unloading forklift operators. A considerable reduction in the total time required to putaway all needed materials can be achieved by increasing the number of forklift operators and adjusting the number of teams formed depending on the percentage of containers that require additional processing.

Chapter 6

Conclusions, Research Contributions and Future Research

In this dissertation, we studied and optimized multiple areas of the inbound supply chain of a manufacturing company. Further, we explained how a manufacturer could cohesively use all optimization models we developed to make better overall operational-level decisions. In Chapter 2, we studied the problem of reducing a manufacturer's long-term expected procurement cost by determining the optimal order quantities to be procured using each available transportation mode under forecast and inventory uncertainty. We solved a two-stage stochastic program using the sample average approximation method and progressive hedging algorithm, and obtained the optimal weekly order quantities to be placed with the overseas suppliers and procured using sea-freight.

Based on the numerical study presented in Chapter 2, we conclude that a good quality solution could be obtained by solving a problem with a large number of scenarios sampled using simple random sampling. However, the computational time required to solve a two-stage stochastic program increases with the number of scenarios. Therefore, we solved our sample approximation problem using the Progressive Hedging Algorithm (PHA). Using the numerical results, we provided managerial insights on the number of scenarios that need to be considered and replications that need to be performed to obtain a good-quality solution in a reasonable amount of time. Additionally, using the value of the stochastic solution, we showed that the procurement cost could be reduced by solving the stochastic program instead of the mean value problem.

We believe that our study in Chapter 2 is the first in the literature to account for the errors in total available inventory. Further, we study a modal splitting problem under inventory and forecast error. Our literature review have shown that the research in the area of modal splitting is underdeveloped. In this study, we compared different sampling methods and different solution methods, such as solving the extensive form of a two-stage stochastic program as is, unbundled progressive hedging algorithm, and bundled progressive hedging algorithm using the performance metrics such as optimality gap, the standard deviation or the estimate of the expected objective value, solution time, sample generation time and convergence rate. We also studied the effects of different levels of error uncertainty on the quality of the solution obtained.

In the literature, there are methods proposed to compute the lower bounds of the PHA. However, the lower bounds obtained using these methods are not always tight. Therefore, future research can be done in this direction to compute tight lower bounds. In section 2.7 of Chapter 2, we propose and discuss a potential approach to achieve this by modifying the non-anticipativity constraints and updating the dual prices using the sub-gradient method.

The orders procured from overseas suppliers using sea-freight are first delivered to the overseas consolidation facility using ground transportation services that perform milk-runs to collect orders from different suppliers. The consolidation facility consolidates the orders from inbound trailers into outbound sea-containers. These sea-containers are then delivered to the overseas port from where they are transported to the port in the USA using sea-freight. The sea-containers that arrive at the USA port are delivered to the manufacturing plant using rail and road transportation services. The parts in these sea-containers are then used to replenish the manufacturer's warehouse inventory to support the production process. Due to consolidation activity, every sea-container contains multiple part types. Further, when a sea-container is unloaded to replenish the warehouse inventory, the manufacturer unloads it completely. Therefore, any unnecessary parts that were packed into the sea-container are also unloaded along with the necessary parts. This results in increased manufacturer's warehouse utilization.

The manufacturing plant has limited space for holding inventories of various parts required for production, and it is crucial to maintain a low warehouse utilization to ensure a smooth production process. In Chapter 3, we assumed that the manufacturer could hold one to two days of parts supply. Therefore, the warehouse will be over-utilized if the entire week's order of a part is consolidated in a single container. In this chapter, we build and study the time-series forecasting models to predict the daily demand values. We propose the forecasting model and the amount of training data to be used by the manufacturer by performing a comprehensive study and evaluation of different models. The numerical study suggests that the resulting predictions can be used to obtain parts daily sea-containers consolidation quantities and reduce the manufacturer's daily warehouse utilization.

Additionally, in Chapter 3, we assumed that the manufacturer practices *Vendor Hub* inventory control for fulfilling the emergency orders that arise due to shortages of a part. Therefore, we formulate a chance-constraint optimization model to determine the optimal emergency stock levels the supplier must maintain under forecast error uncertainty. We did an exhaustive analysis to determine the number of scenarios and significance level of the sample approximation problem to be used by the manufacturer to solve the chance-constraint program. The results obtained by solving the proposed chance-constraint program suggest that a sufficient amount of emergency stock is maintained at the supplier preventing shortages in all thirteen weeks we studied.

In the literature, many studies consider the *traditional* or *vendor managed inventory* approach. Based on the literature we reviewed, we believe that our study is the first to consider the *vendor hub* approach and determine the supplier stock levels under forecast error uncertainty using probabilistic constraints. Using the chance constraint optimization model, we also studied the effects of different levels of forecast error uncertainty on the quality of solutions obtained under different experimental settings.

If the manufacturer uses the approach discussed in Chapter 2 to place the orders with overseas suppliers and needs to consolidate the sea-containers using the daily demand to reduce the warehouse utilization, applying the approach discussed in Chapter 3 to determine the daily consolidation quantities requires the use of hierarchical forecast reconciliation. In hierarchical reconciliation, the top-level is weekly order quantities procured using sea-freight (i.e., first-stage decision variable value), which is determined using the approach in Chapter 2. At the bottom-level are the daily forecasted demand quantities predicted using ensemble forecasting models. The top-level forecast values can be assumed to be fixed as they minimize the overall supply chain costs, and the bottom-level forecast values can be readjusted such that the sum of all bottom-level values is equal to the top level. When a sea-container consolidated using this approach is unloaded to replenish the stock of required material, the warehouse would not be filled with an entire week's supply of necessary and unnecessary parts contained inside the sea-container. This approach can be evaluated as part of future research.

The part mix in the outbound sea-containers from the consolidation center depends on the part mix (as a result of milk runs) in the inbound trailers to the consolidation center. Therefore, a vehicle routing problem and trailer assignments problem in the consolidation center can be solved such that a *favorable* part mix is achieved in the outbound sea-containers, reducing the manufacturer's daily warehouse utilization. This is another potential future research direction.

The manufacturer needs to replenish its warehouse inventory levels daily for multiple parts using the parts received in sea-containers and trailers. Due to the consolidation process, each sea-container and trailer contains a mixture of multiple parts. The manufacturer incurs a daily detention penalty (charged by the freight partners who provide the sea-containers and trailers) for storing the sea-containers and trailers on its property. Thus, the sea-containers and trailers must be unloaded and returned to the freight partner as soon as possible. However, the manufacturer cannot unload all sea-containers and trailers because it has limited warehouse space. In Chapter 4, we formulated and solved a multi-objective optimization model to determine the sea-containers and trailers that need to be unloaded on a given day such that the warehouse utilization and detention penalties are minimized. To the best of our knowledge, our study is the first in the literature to simultaneously consider two objectives: minimizing the number of remnant parts in the warehouse and detention penalty costs. We solved the multi-objective model using different methods and compared their performances using the data obtained from our industrial partner. Based on the numerical results, we proposed the goal programming method to be implemented by the manufacturer for selecting the sea-containers and trailers that need to be unloaded daily. Further, we developed a logic to solve the goal programming problem by systematically relaxing the constraint to obtain a feasible solution in any circumstance.

Once the sea-containers and trailers that need to be unloaded are selected using the approach described in Chapter 4, the pallets of parts in these containers are unloaded and processed by the teams of workers. In Chapter 5, we build and study a simulation optimization model to determine the number of teams and the number of people in each team such that the total expected time required to unload all the required sea-containers and trailers on a day is minimized. The proposed simulation-based optimization model helps alleviate bottlenecks in the process by appropriately forming the teams and assigning the workers to each team. In this chapter, we also studied the effects of the workload on team formation.

In Chapter 5, we assumed that the workers are the same. Future research can be done on team formation and worker assignment by considering the workers to be different. This study also assumed that the scanner only scans the pallets in a given team. We think that the overall expected processing times can be reduced by switching the scanner to the breakdowner whenever a bottleneck is created at the breakdown station. A future study can also be done to compare the results obtained using inflexible and flexible scanners in a team.

Bibliography

Acar, Y. and Gardner Jr, E. S. (2012). Forecasting method selection in a global supply chain. *International Journal of Forecasting*, 28(4):842–848.

Afshar, A. and Haghani, A. (2012). Modeling integrated supply chain logistics in real-time large-scale disaster relief operations. *Socio-Economic Planning Sciences*, 46(4):327–338.

Ahmed, S. and Shapiro, A. (2008). *Solving Chance-Constrained Stochastic Programs via Sampling and Integer Programming*, chapter 12, pages 261–269. INFORMS TutORials in Operations Research.

Ahmed, S., Shapiro, A., and Shapiro, E. (2002). The sample average approximation method for stochastic programs with integer recourse. *Submitted for publication*, pages 1–24.

Andradóttir, S. and Ayhan, H. (2005). Throughput maximization for tandem lines with two stations and flexible servers. *Operations Research*, 53(3):516–531.

Askin, R. G. and Huang, Y. (2001). Forming effective worker teams for cellular manufacturing. *International Journal of Production Research*, 39(11):2431–2451.

Azadi, Z., Eksioglu, S. D., Eksioglu, B., and Palak, G. (2019). Stochastic optimization models for joint pricing and inventory replenishment of perishable products. *Computers & Industrial Engineering*, 127:625–642.

Barbarosoğlu, G. and Arda, Y. (2004). A two-stage stochastic programming framework for transportation planning in disaster response. *Journal of the Operational Research Society*, 55(1):43–53.

Barnes, E., Dai, J., Deng, S., Down, D., Goh, M., LAU, H. C., and Moosa, S. (2003). On the strategy of supply hubs for cost reduction and responsiveness.

Berghman, L. and Leus, R. (2015). Practical solutions for a dock assignment problem with trailer transportation. *European Journal of Operational Research*, 246(3):787–799.

Bidhandi, H. M. and Yusuff, R. M. (2011). Integrated supply chain planning under uncertainty using an improved stochastic approach. *Applied Mathematical Modelling*, 35(6):2618–2630.

Bilsel, R. U. and Ravindran, A. (2011). A multiobjective chance constrained programming model for supplier selection under uncertainty. *Transportation Research Part B: Methodological*, 45(8):1284–1300.

Birge, J. R. and Louveaux, F. (2011). *Introduction to stochastic programming*. Springer Science & Business Media.

Burke, G. J., Carrillo, J. E., and Vakharia, A. J. (2009). Sourcing decisions with stochastic supplier reliability and stochastic demand. *Production and Operations Management*, 18(4):475–484.

Carlo, H. J., Vis, I. F., and Roodbergen, K. J. (2014). Storage yard operations in container terminals: Literature overview, trends, and research directions. *European Journal of Operational Research*, 235(2):412–430.

Carøe, C. C. and Schultz, R. (1999). Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24(1-2):37–45.

Chen, P., Guo, Y., Lim, A., and Rodrigues, B. (2006). Multiple crossdocks with inventory and time windows. *Computers & Operations Research*, 33(1):43–63.

Choi, J., Cao, J. J., Romeijn, H. E., Geunes, J., and Bai, S. X. (2005). A stochastic multi-item inventory model with unequal replenishment intervals and limited warehouse capacity. *IIE Transactions*, 37(12):1129–1141.

Chopra, S. and Meindl, P. (2013). *Supply Chain Management: Strategy, planning, and Operation*. Pearson.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197.

Dehghani, M., Abbasi, B., and Oliveira, F. (2021). Proactive transshipment in the blood supply chain: A stochastic programming approach. *Omega*, 98:102112.

Dillon, M., Oliveira, F., and Abbasi, B. (2017). A two-stage stochastic programming model for inventory management in the blood supply chain. *International Journal of Production Economics*, 187:27–41.

Ekren, B. Y., Evans, G. W., Heragu, S. S., and Usher, J. S. (2012). Simulation-based performance improvement of a defense logistics warehouse. In *12th IMHRC Proceedings*, 18, Gardanne, France.

Engebrethsen, E. and Dauzère-Pérès, S. (2019). Transportation mode selection in inventory models: A literature review. *European Journal of Operational Research*, 279(1):1–25.

Facebook Open Source (2017). Prophet.

Fan, Y., Schwartz, F., and Voß, S. (2017). Flexible supply chain planning based on variable transportation modes. *International Journal of Production Economics*, 183:654–666.

Fisher, M. L. (2004). The lagrangian relaxation method for solving integer programming problems. *Management Science*, 50(12_supplement):1861–1871.

Fowler, J. W., Wirojanagud, P., and Gel, E. S. (2008). Heuristics for workforce planning with worker differences. *European Journal of Operational Research*, 190(3):724–740.

Gade, D., Hackebeil, G., Ryan, S. M., Watson, J.-P., Wets, R. J.-B., and Woodruff, D. L. (2016). Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs. *Mathematical Programming*, 157(1):47–67.

Gamberini, R., Grassi, A., Mora, C., and Rimini, B. (2008). An innovative approach for optimizing warehouse capacity utilization. *International Journal of Logistics*, 11(2):137–165.

Ganbold, O., Kundu, K., Li, H., and Zhang, W. (2020). A simulation-based optimization method for warehouse worker assignment. *Algorithms*, 13(12):326.

Govindan, K. (2015). The optimal replenishment policy for time-varying stochastic demand under vendor managed inventory. *European Journal of Operational Research*, 242(2):402–423.

Gu, J., Goetschalckx, M., and McGinnis, L. F. (2007). Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, 177(1):1–21.

Gu, J., Goetschalckx, M., and McGinnis, L. F. (2010). Research on warehouse design and performance evaluation: A comprehensive review. *European Journal of Operational Research*, 203(3):539–549.

Guo, L., Fang, W., Zhao, Q., and Wang, X. (2021). The hybrid prophet-svr approach for forecasting product time series demand with seasonality. *Computers & Industrial Engineering*, 161:107598.

Hamdan, B. and Diabat, A. (2019). A two-stage multi-echelon stochastic blood supply chain problem. *Computers & Operations Research*, 101:130–143.

Hanbazazah, A. S., Abril, L., Erkoc, M., and Shaikh, N. (2019). Freight consolidation with divisible shipments, delivery time windows, and piecewise transportation costs. *European Journal of Operational Research*, 276(1):187–201.

Hanbazazah, A. S., Castro, L. E., Erkoc, M., and Shaikh, N. I. (2020). In-transit freight consolidation of indivisible shipments. *Journal of the Operational Research Society*, 71(1):37–54.

Heragu, S. S., Du, L., Mantel, R. J., and Schuur, P. C. (2005). Mathematical model for warehouse design and product allocation. *International Journal of Production Research*, 43(2):327–338.

Hyndman, R. J. and Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

Hyndman, R. J. and Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for r. *Journal of Statistical Software*, 27:1–22.

Işık, T., Andradóttir, S., and Ayhan, H. (2016). Optimal control of queueing systems with non-collaborating servers. *Queueing Systems*, 84(1):79–110.

Jalali, H. and Nieuwenhuyse, I. V. (2015). Simulation optimization in inventory replenishment: a classification. *IIE Transactions*, 47(11):1217–1235.

Kleywegt, A. J., Shapiro, A., and Homem-de Mello, T. (2002). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502.

Küçükyavuz, S. and Sen, S. (2017). An introduction to two-stage stochastic mixed-integer programming. In *Leading Developments from INFORMS Communities*, pages 1–27. INFORMS.

Kutzelnigg, R. (2011). Optimal allocation of goods in a warehouse: Minimizing the order picking costs under real-life constraints. In *3rd IEEE International Symposium on Logistics and Industrial Informatics*, pages 65–70. IEEE.

Lee, C. C. and Chu, W. H. J. (2005). Who should control inventory in a supply chain? *European Journal of Operational Research*, 164(1):158–172.

Li, G., Yang, S., Xu, Z., Wang, J., Ren, Z., and Li, G. (2020). Resource allocation methodology based on object-oriented discrete event simulation: A production logistics system case study. *CIRP Journal of Manufacturing Science and Technology*.

Li, N., Chiang, F., Down, D. G., and Heddle, N. M. (2021). A decision integration strategy for short-term demand forecasting and ordering for red blood cell components. *Operations Research for Health Care*, 29:100290.

Lin, Y.-S. and Wang, K.-J. (2018). A two-stage stochastic optimization model for warehouse configuration and inventory policy of deteriorating items. *Computers & Industrial Engineering*, 120:83–93.

Linderoth, J., Shapiro, A., and Wright, S. (2006). The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research*, 142(1):215–241.

Lucidi, S., Maurici, M., Paulon, L., Rinaldi, F., and Roma, M. (2016). A simulation-based multiobjective optimization approach for health care service management. *IEEE Transactions on Automation Science and Engineering*, 13(4):1480–1491.

Luedtke, J. and Ahmed, S. (2008). A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19(2):674–699.

Mak, W.-K., Morton, D. P., and Wood, R. K. (1999). Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters*, 24(1-2):47–56.

McKay, M. D., Beckman, R. J., and Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61.

Miao, Z., Lim, A., and Ma, H. (2009). Truck dock assignment problem with operational time constraint within crossdocks. *European Journal of Operational Research*, 192(1):105–115.

Moheb-Alizadeh, H. and Handfield, R. (2018). An integrated chance-constrained stochastic model for efficient and sustainable supplier selection and order allocation. *International Journal of Production Research*, 56(21):6890–6916.

Mosca, A., Vidyarthi, N., and Satir, A. (2019). Integrated transportation–inventory models: A review. *Operations Research Perspectives*, 6:100101.

Motamedi, M., Li, N., Down, D. G., and Heddle, N. M. (2021). Demand forecasting for platelet usage: from univariate time series to multivariate models. *arXiv preprint arXiv:2101.02305*.

Muriel, A. and Ruiz-Benitez, R. (2006). Optimal implementation and benefits of rolling inventory. *IIE Transactions*, 38(7):591–603.

Negahban, A. and Smith, J. S. (2014). Simulation for manufacturing system design and operation: Literature review and analysis. *Journal of Manufacturing Systems*, 33(2):241–261.

Pauls-Worm, K. G., Hendrix, E. M., Alcoba, A. G., and Haijema, R. (2016). Order quantities for perishable inventory control with non-stationary demand and a fill rate constraint. *International Journal of Production Economics*, 181:238–246.

Poon, T., Choy, K. L., Chan, F. T., Ho, G. T., Gunasekaran, A., Lau, H. C., and Chow, H. K. (2011). A real-time warehouse operations planning system for small batch replenishment problems in production environment. *Expert Systems with Applications*, 38(7):8524–8537.

Poormoaied, S., Atan, Z., de Kok, T., and Van Woensel, T. (2020). Optimal inventory and timing decisions for emergency shipments. *IISE Transactions*, 52(8):904–925.

Poormoaied, S., Atan, Z., and van Woensel, T. (2022). Quantity-based emergency shipment policies. *IISE Transactions*, pages 1–13.

Poulos, P., Rigatos, G., Tzafestas, S., and Koukos, A. (2001). A pareto-optimal genetic algorithm for warehouse multi-objective optimization. *Engineering Applications of Artificial Intelligence*, 14(6):737–749.

Rawls, C. G. and Turnquist, M. A. (2012). Pre-positioning and dynamic delivery planning for short-term response following a natural disaster. *Socio-Economic Planning Sciences*, 46(1):46–54.

Rockafellar, R. T. and Wets, R. J.-B. (1991). Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147.

Salas, L. C., Cárdenas, M. R., and Zhang, M. (2012). Inventory policies for humanitarian aid during hurricanes. *Socio-Economic Planning Sciences*, 46(4):272–280.

Sarno, D. and Nenni, M. E. (2016). Daily nurse requirements planning based on simulation of patient flows. *Flexible Services and Manufacturing Journal*, 28(3):526–549.

Sonntag, D., Schrotenboer, A., and Kiesmüller, G. P. (2021). Stochastic inventory routing with time-based shipment consolidation. *Transportation Research. Part B: Methodological*, 20(X).

Tolooie, A., Maity, M., and Sinha, A. K. (2020). A two-stage stochastic mixed-integer program for reliable supply chain network design under uncertain disruptions and demand. *Computers & Industrial Engineering*, 148:106722.

Tsui, L. Y. and Chang, C.-H. (1992). An optimal solution to a dock door assignment problem. *Computers & Industrial Engineering*, 23(1-4):283–286.

Tyan, J. C., Wang, F.-K., and Du, T. C. (2003). An evaluation of freight consolidation policies in global third party logistics. *Omega*, 31(1):55–62.

Van Oyen, M. P., Gel, E. G., and Hopp, W. J. (2001). Performance opportunity for workforce agility in collaborative and noncollaborative work systems. *IIE Transactions*, 33(9):761–777.

Watson, J.-P. and Woodruff, D. L. (2011). Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computational Management Science*, 8(4):355–370.

Weng, S.-J., Cheng, B.-C., Kwong, S. T., Wang, L.-M., and Chang, C.-Y. (2011). Simulation optimization for emergency department resources allocation. In *Proceedings of the 2011 winter simulation conference (WSC)*, pages 1231–1238. IEEE.

Wickramasuriya, S. L., Athanasopoulos, G., and Hyndman, R. J. (2019). Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association*, 114(526):804–819.

Wirojanagud, P., Gel, E. S., Fowler, J. W., and Cardy, R. (2007). Modelling inherent worker differences for workforce planning. *International Journal of Production Research*, 45(3):525–553.

You, F., Wassick, J. M., and Grossmann, I. E. (2009). Risk management for a global supply chain planning under uncertainty: models and algorithms. *AIChE Journal*, 55(4):931–946.

Zhang, J.-H., Li, J., and Liu, Z.-P. (2012). Multiple-resource and multiple-depot emergency response problem considering secondary disasters. *Expert Systems with Applications*, 39(12):11066–11071.

Zhang, Y., Lin, W.-H., Huang, M., and Hu, X. (2021). Multi-warehouse package consolidation for split orders in online retailing. *European Journal of Operational Research*, 289(3):1040–1055.