A Multi-period supply planning problem for managing stochastic demand

Shurui Chen

A Thesis

in

The Department

of

Supply Chain and Business Technology Management

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Supply Chain Management (MSCM) at

Concordia University

Montréal, Québec, Canada

August 2024

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Shurui Chen**

Entitled: **A Multi-period supply planning problem for managing stochastic demand**

and submitted in partial fulfillment of the requirements for the degree of

Master of Supply Chain Management (MSCM)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

______________________________ Chair
Dr. Navneet Vidyarthi

______________________________ External Examiner
Dr. Navneet Vidyarthi

______________________________ Examiner
Dr. Ahmet Satir

______________________________ Supervisor
Dr. Satyaveer S. Chauhan

Approved by ____
Rustam Vahidov, Chair
Department of Supply Chain and Business Technology Management

____________ 2024 ______________________________
Anne-Marie Croteau, Dean
John Molson School of Business

Abstract

A Multi-period supply planning problem for managing stochastic demand

Shurui Chen

In this thesis, we address the critical issue of drug shortages that has persisted during and after the Covid-19 pandemic. We develop a two-stage stochastic model aimed at understanding how a compounding pharmaceutical company can balance supplier selection, drug production, back orders, and inventory management when faced with uncertain demand. To explore and resolve the application of two models under varying conditions, we introduce the Sample Average Approximation (SAA) and Lagrangean Decomposition methods. Additionally, to better simulate the uncertainties present in real-world scenarios, we employ Monte Carlo simulations to generate diverse cases, enabling the model to produce more reliable results.

Keywords: Covid-19 Pandemic Drug Shortage, Two-stage stochastic model, Sample Average Approximation (SAA), Lagrangean Decomposition, Monte Carlo simulations

Acknowledgments

Before beginning this thesis, I would like to express my deepest gratitude to my professor, Dr. Satyaveer S. Chauhan, who provided invaluable assistance throughout the process of writing this thesis. His guidance was instrumental in designing the research problem, and under his supervision, I was able to complete the methodology and writing. It was also through Dr. Chauhan's courses at Concordia University that I developed a deep understanding and interest in supply chain design and optimization.

I would also like to extend my sincere thanks to all the professors at Concordia University who have taught me throughout my studies. Their guidance has been instrumental in teaching me how to build models and how to critically read and analyze extensive academic literature.

Finally, I would like to thank my parents for their unwavering financial and emotional support, enabling me to complete my master's degree at Concordia University.

Contents

List of Figures **vii**

List of Tables **viii**

1 Introduction **1**

2 Literature Review **5**

2.1 Background Overview 5

2.2 Mainstream Method and Algorithms 6

2.3 Scenarios Generation Algorithm 8

2.4 Contribution of the Paper 10

3 Mathematical Formulation **11**

3.1 Introduction of the Background and Model 11

3.2 Stochastic Uncertain Demand Problems for Inventory Optimization 12

3.3 Stochastic Uncertain Demand for Model 1 14

3.4 Stochastic Uncertain Demand for Model 2 15

4 Sample Average Approximation **17**

4.1 Introduction of the Sample Average Approximation 17

4.2 Application of the Sample Average Approximation 18

4.3 SAA Algorithm in Model 1 20

5 Lagrangean based Stochastic Decomposition Approach 23
5.1 Introduction of Lagrangean Decomposition 23
5.2 Application on the Model 2 24
5.3 Application of the Algorithm 25
5.4 Algorithm of Solving Lagrangean Dual 27
5.5 Upper Bound Procedure 29

6 Computation Experiment 31
6.1 Convergence of the Algorithm Sample Average Approximation 31
6.2 Experiment for the Small to Medium Size Problem for Model 1 34
6.3 Experiment for the Large Scale Problem for Model 1 36
6.4 Sample Average Approximation Method Application for Model 2 37
6.5 Best Combination for K and S for Model 2 37
6.6 Computational Results for Different Scales Under Model 2 38
6.7 Convergence of Lagrangean Decomposition for Model 2 40
6.8 Test for Small to Medium Scale Problem 40
6.9 Test for Large Scale Problem 41
6.10 Conclusion 42
6.10.1 Problem Overview and Research Conclusions 42
6.10.2 Limitations and Future Research Directions 43

Appendix A My Appendix 45
A.1 Two-Stage Stochastic Model Code 45

Bibliography 46

List of Figures

Figure 5.1 Lagrangean Decomposition Algorithm Flowchart 28

Figure 6.1 SAA Model 1 Percentage of Gap 33

Figure 6.2 SAA Model 1 Calculation Time 33

Figure 6.3 SAA Model 2 Percentage of Gap 38

Figure 6.4 SAA Model 2 Computation Time 38

List of Tables

Table 6.1 SAA Model 1 for Small and Medium Scale Problem 35

Table 6.2 SAA Model 1 for Large Scale Problem 36

Table 6.3 SAA Model 2 for Small to Medium Scale Problem 39

Table 6.4 SAA Model 2 for Large Scale Problem 39

Table 6.5 Lagrangean Decomposition for Small and Medium Scale Problem 41

Table 6.6 Lagrangean Decomposition for Large Scale Problem 42

Chapter 1

Introduction

During 2020, the outbreak of COVID caused the breakdown of the supply chain of medical industry all around the world. The shortage of high-demand items such as personal protective equipment (PPE), antibiotic medicine, and other active pharmaceutical ingredients (API) exacerbated the challenges faced by governments around the world in controlling the pandemic (Magableh, 2021). The price of the API increased two times or even more during and after the pandemic. In the two to three years following the pandemic, especially during each year's flu season, medical shortages have become a very common and significant problem for the healthcare system. Beginning in the summer of 2022 and continuing into early 2023, Canada experienced a significant shortage of medicine used to cure children's flu. The shortage of 2700 types of drug were caused by the manufacturing problem, shipping delay, and unexpected increase in demand. The consequences of the shortage of medicine lead to the patients 'commonly reported to have increased out of pocket costs, rates of drug errors, adverse events, complaints and mortality during times of shortage.

Due to drug shortages, the FDA issued guidance during the COVID-19 pandemic. This guidance permits outsourcing facilities registered with the FDA under section 503B and compounding pharmaceutical companies under section 503A to compound drugs. Specifically, 503B outsourcing facilities can compound essentially copies of approved drugs or use bulk substances not on the FDA's approved list for hospitalized COVID-19 patients during supply disruptions. Similarly, 503A pharmacies can compound shortage drugs without patient prescriptions for hospital use, provided the conditions in the guidance are met (Gianturco, Yoon, Yuen, & Mattingly, 2021). This policy has made compounding pharmaceutical companies an important part of the drug supply chain.

Compounding pharmaceutical companies specialize in creating customized medications tailored to the specific needs of patients. The operation process of these companies typically involves purchasing, packing, and selling Active Pharmaceutical Ingredients (APIs), usually processed in-house or through third-party outsourcing. This approach allows them to produce a wide variety of formulations and dosages that are not available from standard pharmaceutical manufacturers.

Due to their limited storage capacities, these companies often adopt just-in-time (JIT) inventory methods and rely heavily on a network of outsourced suppliers to fulfill production needs. Their demand can be highly variable, influenced by seasonal trends and specific patient needs. Without the large-scale warehousing capabilities of major pharmaceutical companies, compounding pharmaceutical companies must efficiently manage their supply chains to ensure timely production and delivery.

Previous studies have explored inventory routing optimization in the pharmaceutical industry by setting the hub location when facing the demand uncertainty. Nikzad, Bashiri, and Oliveira (2019) developed models for pharmaceutical supply chain inventory routing management, considering factors such as shortage, demand, and transportation cost. While those models offer valuable insights, researchers often assume the availability of large storage and production capacities. However, most small to mid-sized compounding pharmaceutical companies operate with limited storage space and financial resources, making these models less applicable to their circumstances. Therefore, the optimization challenge faced by compounding pharmaceutical companies can be characterized as an inventory optimization model with multiple constraints and uncertain demand.

The predominant methodologies for addressing that kind of problem are stochastic programming and robust optimization Contreras, Cordeau, and Laporte (2011a). The most notable distinction between these two methods comes from the distribution of the uncertain parameters. In stochastic programming, the uncertainty is governed by known probability distributions, while in robust optimization, the uncertainty is considered completely random without any known distribution. Given the availability of extensive historical data in the pharmaceutical industry, this study will adopt the stochastic model to address the inventory optimization challenges. The stochastic programming separates to two predominant categories, two stage stochastic programming and chance constraint (Nikzad et al., 2019). Two-stage stochastic programming proposed by Dantzig (1955) minimizes the expected value of the recourse function, while chance-constrained programming developed by Charnes and Cooper (1963) provides solutions for scenarios where constraints might be violated.

Due to the influence of various factors on the uncertainties in real-world problems, generating scenarios corresponding to each demand is a highly challenging research direction. As a result, many researchers use different methods to generate scenarios that simulate real-world uncertainties. Common approaches include Monte Carlo simulations, Latin Hypercube Sampling (LHS), Markov processes, and dynamic programming (Adelman, 2004). Given the large number of samples we generated and the widespread popularity of Monte Carlo simulation, we will employ the Mont Carlo to generate the scenarios.

We designed our model based on operational and procurement practices of compounding pharmaceutical company that prefers Just-In-Time (JIT) model and relies heavily on outsourcing. We assume, the company do not want to keep any safety stock or maintain only the contingency supply. In this condition, we want make sure that the supply chain must pick a right set of suppliers capable of supplying varying quantity of material in each period during the planning horizon. We also assume that each supplier expect a minimum business as a part of supply chain, so we included this fact in pour model i.e. if a supplier is selected, supply chain will source a minimum pre-agreed quantity during the planning horizon. In addition, we assume that this a limited planning horizon problem where we produce products for a seasonal demand and we assume that the product cannot be inventoried for the next season - e.g. seasonal flu vaccine.

Since, the demand is stochastic and planning horizon is limited, our overall problem to select suppliers capable of managing the demand in the most economical way. The first stage of the model involves selecting multiple suppliers for specific products, while in the second stage, the capacities of these suppliers are leveraged to address uncertain demand, aiming to avoid inventory overstock and replenishment pressures. The objective is to minimize the overall procurement and other inventory related costs. Due to the multiple demand scenarios and integrity of some decision variables, we explore simulation and decomposition based techniques. To refine our model for diverse operational environments, we applied the Sample Average Approximation (SAA) for simulation calculations. To assess SAA's adaptability, we compared it with Lagrangian decomposition approach. This comparison helps us choose the most effective method for inventory management across different scenarios, enhancing the model's flexibility and reliability.

The organization of the paper is structured as follows: Chapter 2 provides a review of the relevant literature. Chapter 3 describes the problem and outlines the mathematical models used. Chapter 4 discusses the mathematical methods for solving the proposed model and the scenario generation techniques. Chapter 5 presents Lagrangean Decomposition. Finally, results and conclusions are drawn in chapter 6.

Chapter 2

Literature Review

2.1 Background Overview

In the pharmaceutical supply chain, limited production capacity and surges in demand are frequent issues. Since the outbreak of COVID-19, these challenges have often been highlighted in popular media as drug shortages.Outsourcing facilities support public health institutions in two primary ways. First, they provide ready-to-use sterile drug products by compounding on a larger scale without requiring patient-specific prescriptions. Second, they ensure the timely availability of drug products during shortages, bridging gaps when manufacturers discontinue products or other supply issues arise (Gianturco et al., 2021).

Compounding pharmaceutical companies, often serving as a key outsourcing resource, typically employ the Just-In-Time (JIT) operating model due to their requirement for high turnover rates and limited storage space. JIT is an innovative inventory management strategy that promotes continuous production flow and aims to accurately fulfill market demand (Baum, 2006). In a JIT system, suppliers deliver stock precisely when it is needed, minimizing inventory costs by decreasing the amount of unused inventory (Aptel & Pourjalali, 2001). However, this method brings both benefits and drawbacks for compounding pharmaceutical companies. A major issue is the unpredictability and variability of hospital workload volumes. This uncertainty can pose a serious threat to hospital operations if demand suddenly spikes and the available inventory is insufficient (Magableh, 2021). Balkhi, Alshahrani, and Khan (2022) proposed solutions for addressing these challenges include:

maintaining a buffer inventory to mitigate supply disruptions, offering customers medical products that are not directly related to patient care, and enhancing flexibility in supplier relationships to ensure rapid response to changing demands.

Given our current situation of limited storage space, the option of maintaining a buffer inventory is not the most optimal choice. Additionally, supplying APIs is directly linked to patient health and safety. Therefore, flexible supplier selection will be a critical aspect in the development of our model.

2.2 Mainstream Method and Algorithms

Compounding pharmaceutical companies will initially identify and select multiple backup suppliers; however, in the subsequent stage, they will rely on these suppliers to alleviate inventory pressure and effectively meet market demand. This issue has transformed into an unanticipated supplier selection with fluctuated demand problem. However, to the best of our knowledge, majority of current mainstream literature focuses on unanticipated hub location with multiple assignments or unanticipated inventory routing problems, although not identical, the research objectives share similarities, providing a methodological foundation for our study.

In most studies related to the inventory routing problem (IRP) with uncertain demand, researchers focus on two stage stochastic programming, which integrates two components: inventory management and vehicle routing. These two problems are traditionally considered separately; however, the IRP combines them together (Campbell & Savelsbergh, 2004).

A specific category of product distributed by homogeneous fleets (Abdelmaguid, Dessouky, & Ordóñez, 2009) is the main stream. (Li, Zhang, & Zhang, 2020) introduced two-stage stochastic model in solving the IRP problem for optimizing the refined oil distribute problem, of which the uncertainty comes from the sales volume in each hub. Alinaghian and Zamani (2019) proposed a bi-objective model for the green inventory routing problem, focusing on emissions reduction in first stage, while fleet size, inventory costs, and routing costs minimization in second stage. Coelho, Cordeau, and Laporte (2012) presented an IRP model that incorporates heterogeneous fleets and proposed a heuristic mathematical strategy for problem resolution.

As research progresses and the complexity of practical problems escalates, an increasing number of scholars are focusing on the challenges associated with managing diverse products within heterogeneous fleets and warehouse environments. This area of study addresses the intricacies of deploying multiple vehicle types alongside the strategic alignment of varied product needs and differing storage capabilities. Ramadhan, Imran, and Rizana (2018) tackle a multi-product, multi-period, and multi-supplier inventory routing problem (IRP) using a record-to-record travel algorithm. This method seeks optimal solutions rapidly for scenarios where a fleet of homogeneous vehicles transports products from suppliers to an assembly plant within a finite time horizon. Moin (2011) optimized a multi-period and multi-product IRP model with time-varying demand, using a variable neighborhood search algorithm. Mirzapour Al-e Hashem and Rekik (2014) considered the problem addresses a multi-product, multi-period Inventory Routing Problem (IRP) involving the distribution of products by heterogeneous vehicles. These studies collectively push the boundaries of IRP, addressing both environmental sustainability and the logistical demands of diverse product types and transport modalities.

Their algorithm can be categorized into three types based on the timing of demand disclosure and the nature of demand itself, encompassing static, stochastic, and dynamic approaches. The static and stochastic methods rely on demands with known distributions, whereas the dynamic approach handles demands whose distributions are not predetermined (Nikzad et al., 2019).

Stochastic programming can be divided into two categories based on the techniques used to represent uncertainty(Nikzad et al., 2019). The first category involves determining a first-stage strategy, followed by constraining and optimizing the second stage. This approach seeks to optimize the objective function without violating any constraints. Azadeh, Elahi, Farahani, and Nasirian (2017) employs a genetic algorithm-Taguchi method to address the inventory routing problem for perishable products with transshipment, under uncertain demand conditions. This approach focuses on optimizing the overall system by minimizing the expected value of inventory and transportation costs.

In the first category, further subdivision can be made based on their planning horizon into problems with finite and infinite time horizons. Avella, Boccia, and Wolsey (2018) introduced an innovative approach in their paper ”Single-Period Cutting Planes for Inventory Routing Problems.” They proposed a set of cutting planes derived from a single-period substructure within the context of the Inventory Routing Problem (IRP), focusing on multi-product distribution. Jaillet, Bard, Huang, and Dror (2002) develop a comprehensive method to approximate delivery costs for inventory routing problems, focusing on multi-product distribution over a rolling horizon. Adelman (2004) addressed an infinite inventory-routing problem (IRP) combining inventory management with vehicle routing. He employed a price-directed decomposition method, using optimal dual prices from linear programming relaxations to dynamically generate vehicle itineraries.

Another category is called chance constraints, in which the solutions are calculated with a certain probability of violating the constraints Yu, Chu, Chen, and Chu (2012). A recent study by (Solyalı & Süral, 2017) demonstrates how chance constraints can be applied when studying multi-phase and multi-product problems in inventory optimization. In this paper, the service ratio is leveraged as the chance constraint which can be violated. Nikzad et al. (2019) proposed optimizing the inventory management problem under normally distributed demand using chance constraint methods.

2.3 Scenarios Generation Algorithm

In best of our knowledge, we found that many articles and researchers have employed several mainstream methods to generate the scenarios. The most commonly used methods include Sample Average Approximation (SAA), moment matching, methods based on probability metrics, and Voronoi cell sampling methods (Nikzad et al., 2019). Sample Average Approximation (SAA) is a method that replaces the expected recourse cost function with a sample average, and the corresponding optimization problem is solved using specialized algorithms for non-convex optimization. In the first stage, a sample average approximation of the objective function is created. In the second stage, the optimization problem is solved, and it has been shown that as the sample size increases, the SAA solution converges to the true problem solution (Ahmed, Shapiro, & Shapiro, 2002). Kleywegt, Shapiro, and Homem-de Mello (2002) proposed using the Monte-Carlo simulation as the foundation of sample average approximation to generate the samples and scenarios to simulate the uncertainty. Linhares Rodrigues (2019) applied the sample average approximation to distributional robust stochastic optimization, showing its potential for reducing problems to those with a polynomial-size support. These studies collectively highlight the versatility and effectiveness of the sample average approximation method in addressing various challenges in two-stage stochastic models. Contreras et al. (2011a) proposed to use the sample average approximation method in solving the unanticipated hub location problems with multiple assignments. In Contreras et al. (2011a)'s research, he found that one shortcoming of the sample average approximation is the computational pressure. Therefore, Benders decomposition is used to relieve the computational burden. Linhares Rodrigues (2019) identified a key limitation of the Sample Average Approximation (SAA) method: while effective for generating solutions, it struggles with accuracy and efficiency when dealing with large sample sizes and high coefficients of variation. To address these issues, they proposed LP-rounding algorithms.

Other than sample average approximation method, some researchers also advocate using the scenario decomposition method in solving the two stage stochastic problem. The scenario decomposition method is specially designed for problem of which the first stage contain binary variables (Ahmed, 2013). In Ahmed (2013)'s research paper, the scenario decomposition method is described as an approach first decomposes the complex stochastic programming problem into multiple sub problems, each corresponding to a different scenario. Using the Lagrangian dual problem, the scenario sub problems are solved independently to evaluate the dual values for each scenario. The solutions to these sub problems are then combined to form an approximate solution to the original problem. The algorithm operates by solving scenario sub problems to generate candidate solutions and lower bounds. By evaluating these candidate solutions, the algorithm derives upper bounds for the original problem. (Crainic, Fu, Gendreau, Rei, & Wallace, 2011) proposed a scenario decomposition heuristic framework for solving the stochastic CMND problem, using progressive hedging to efficiently address uncertain demands and optimize network design costs.

Based on our review of the current literature and the challenges we face, we plan to design a two-stage stochastic model utilizing both Sample Average Approximation (SAA) and Lagrangian decomposition methods. For the SAA method, we will employ Monte Carlo simulation to generate various scenarios and samples. As Nikzad et al. (2019) mentioned, ensuring computational accuracy with Monte Carlo simulation requires generating a large number of samples. In addition to SAA, we will also employ scenario decomposition to tackle the stochastic problem. The objective of using these two methods is to perform a comparative analysis, allowing us to gain a more comprehensive understanding and to determine the most suitable approach for evaluating problems and models under different conditions in the future.

2.4 Contribution of the Paper

Different from previously mentioned Inventory Routing Problem (IRP) studies, our research focuses on supplier selection rather than hub allocation and omits the traditional routing problem due to the logistical flexibility of compounding pharmaceutical companies. These companies prioritize supply quantities, back orders, and inventory levels. A specific characteristic of compounding pharmaceutical companies is their homogeneous product, enabling us to conduct a multi-period analysis for single products with heterogeneous suppliers.

This paper provides three key contributions based on the current literature review:

(1) **Novel Perspective on Optimization**: We explore a new perspective by optimizing across different stages of production, inventory, and sales. Additionally, we introduce significant flexibility in adjusting the variables included in both the first and second stages, enhancing the model's adaptability to various scenarios.

(2) **Methodological Innovation**: We employ multiple methods, including Sample Average Approximation (SAA) and scenario decomposition. Instead of solely optimizing the computational aspects of the SAA algorithm, we introduce an alternative method for cross-validation, enhancing the robustness and reliability of our results.

(3) **Comparative Analysis**: We conduct a thorough comparative analysis of the two methods across various dimensions, including computational time, accuracy, and applicability. This comprehensive evaluation provides valuable insights into the strengths and limitations of each approach, guiding future research and practical applications.

Chapter 3

Mathematical Formulation

3.1 Introduction of the Background and Model

In our study, we focus on the production and inventory optimization of a specific product over multiple periods, addressing the operational challenges faced by compounding pharmaceutical companies that handle a variety of products. Given the homogeneity of the products in terms of market demand, our analysis prioritizes one type of product to optimize the production and inventory processes effectively. We assume that there are number of suppliers, each capable of producing the designated product. Due to capacity constraints, each production line has a fixed maximum output. The suppliers are heterogeneous, meaning the cost per unit of product and the initial setup cost vary among suppliers. This diversity necessitates a strategic approach to production planning and supplier selection. To align closely with market demands and minimize the risk of stock outs, we incorporate a ’beta service level’ in our model. This parameter ensures that the shortage quantity remains within a specified threshold, thereby maintaining a balance between production efficiency and market responsiveness. This approach not only enhances operational efficiency but also aims to meet customer demand reliably, which is crucial in the pharmaceutical industry where timely supply of products can be critical.

3.2 Stochastic Uncertain Demand Problems for Inventory Optimization

- **Decision Variable**

- Z_i Binary variable represents whether supplier i is selected, for supplier $i = 1 \ldots M$.

- X_{it} Supplier production quantity of product i in period t.

- B_t Back order quantity in period t.

- I_t Inventory quantity in period t.

- **Parameters of the Model**

- f_i Contract cost for selecting supplier i, for $i = 1 \ldots M$.

- ρ_i Cost of producing one unit of product by specific supplier i.

- D_t Uncertain demand for the product in time period t, $t = 1, \ldots, T$.

- α Back ordering cost per unit per period.

- β Inventory holding cost per unit per period.

- C_i Maximum capacity of supplier i.

- MOQ_i Minimum quantity ordered from supplier i.

- γ Gamma service level each period.

Objective function of Model

$$\text{Minimize} \quad \sum_{i=1}^{M} f_i \cdot Z_i + \alpha \cdot \sum_{t=1}^{T} B_t + \beta \cdot \sum_{t=1}^{T} I_t + \rho \cdot \sum_{i=1}^{M} \sum_{t=1}^{T} X_{it} \tag{1}$$

Subject to

$$X_{it} \leq Z_i \cdot C_i \tag{2}$$

$$\sum_{t=1}^{T} X_{it} \geq Z_i \cdot MOQ_i \tag{3}$$

$$I_t = I_{t-1} + \sum_{i=1}^{M} X_{i,t} - D_t - B_{t-1} + B_t \tag{4}$$

$$\sum_{i=1}^{M} \sum_{t=1}^{T} X_{it} \cdot Z_i \geq 0.8 \cdot \sum_{t=1}^{T} D_t \tag{5}$$

$$Z_i \in \{0, 1\} \; \forall i \in M \tag{6}$$

Model Design and Explanation

Assumption

Based on real-world experiences and cases, the assumptions of our model are outlined below:

(1) **Single Product Focus**: Although compounding pharmaceutical companies offer a diverse range of products, our model focuses solely on one product. This is to analyze decisions for a product facing demand changes over multiple periods and extend these findings to other product lines.

(2) **Annual Cycle Assumption**: We assume a one-year demand and supply cycle for the product.

(3) **Multiple Suppliers with Constraints**: Given the broad applicability of the product, we assume multiple suppliers, not limited to just one. Each supplier is subject to capacity and minimum order quantity (MOQ) constraints.

(4) **Service Level Assumption**: To prevent substantial stockouts that could obstruct sales, we assume a certain gamma service level in our model.

Objective Function

The objective function of the model explores the total costs incurred in the production process for a pharmaceutical company. This includes the fixed contract costs associated with selecting suppliers, with Z_i as a binary variable representing whether a supplier has been selected, and f_i as the contract cost. The second part of the function addresses back order costs, with α representing the penalty cost per unit due to lost demand. The third component accounts for inventory costs,with β represents the storage cost per unit, and the final part quantifies the production costs for all quantities produced in each period. This comprehensive approach ensures a detailed financial analysis of the production operations.

Constraints

The model includes several constraints: Constraint (2) ensures that the production quantity from any selected supplier does not exceed their capacity; Constraint (3) requires that the total production quantity from any supplier meets at least the minimum order quantity if selected; Constraint (4) maintains inventory balance by equating the inventory at any period to the previous period's inventory, adjusted by received shipments, demand, and net back orders from the current and previous periods; and Constraint (5) controls the back order ratio to not exceed the predefined service level, ensuring service quality is maintained relative to demand variability.

3.3 Stochastic Uncertain Demand for Model 1

We now consider that the demand in each period for product are stochastic. According to our assumption that the demand are independent random variables, each following a known probability distribution. In the experiment, we will also set the demand as stochastic following the normal distribution. We consider the distribution function of $D_t(\xi)$ can be computed. In our context, this uncertainty in demand directly influences both the back order level, B_t, and the inventory level, I_t. The formula can be stated as below:

$$\text{Minimize} \quad \sum_{i=1}^{M} f_i \cdot Z_i + \rho \cdot \sum_{i=1}^{M}\sum_{t=1}^{T} X_{it} + E_\xi \left(\alpha \cdot \sum_{t=1}^{T} B_t(\xi) + \beta \cdot \sum_{t=1}^{T} I_t(\xi) \right) \tag{7}$$

$$\text{Subject to} \quad X_{it}(\xi) \leq Z_i \cdot C_i, \quad \forall i \in M, \forall t \in T, \forall \xi \in \Xi \tag{8}$$

$$\sum_{t=1}^{T} X_{it}(\xi) \geq Z_i \cdot \text{MOQ}_i, \quad \forall i \in M, \forall \xi \in \Xi \tag{9}$$

$$I_t(\xi) = I_{t-1}(\xi) + \sum_{i=1}^{M} X_{it}(\xi) - D_t(\xi) - B_{t-1}(\xi) + B_t(\xi), \quad \forall t \in T, \forall \xi \in \Xi \tag{10}$$

$$\sum_{i=1}^{M}\sum_{t=1}^{T} X_{it}(\xi) \geq 0.8 \cdot \sum_{t=1}^{T} D_t(\xi), \quad \forall \xi \in \Xi \tag{11}$$

$$Z_i \in \{0,1\} \qquad \forall i \in M \tag{12}$$

Here, E_ξ denotes the expected value with respect to the random variable ξ, and Ξ represents the set of all possible values (support) that ξ can assume. The first component of the model accounts for the setup cost for suppliers and the production cost for outsourcing. In contrast, the second component encompasses the back order and inventory expenses under varying demand scenarios. According to constraints (9) and (10), the back order and inventory levels adjust to meet the required service ratio under different demand conditions. Moreover, previous inventory is utilized to satisfy new demand during sales spikes; otherwise, the back order levels would increase. Therefore, all variables fluctuate with demand uncertainty, making the use of expected values impractical in this context. Contrary to other research papers which use the expected value to represent uncertainty, this model explicitly accounts for the variability in demand.

3.4 Stochastic Uncertain Demand for Model 2

In the previous model one, we set the suppliers selected and the quantity produced as the first stage variables, while left the inventory and back order quantity as the second stage. However, instead of solving the aforementioned problem directly, we modify the model to introduce greater complexity in the second stage. This adjustment allows us to more effectively evaluate the performance and robustness of algorithm we will use in the following chapters.

$$\text{Minimize} \quad \sum_{i=1}^{M} f_i \cdot Z_i + E_\xi \left(\rho \cdot \sum_{i=1}^{M} \sum_{t=1}^{T} X_{it}(\xi) + \alpha \cdot \sum_{t=1}^{T} B_t(\xi) + \beta \cdot \sum_{t=1}^{T} I_t(\xi) \right) \tag{13}$$

$$\text{Subject to} \quad X_{it}(\xi) \leq Z_i \cdot C_i, \quad \forall i \in M, \forall t \in T, \forall \xi \in \Xi \tag{14}$$

$$\sum_{t=1}^{T} X_{it}(\xi) \geq Z_i \cdot \text{MOQ}_i, \quad \forall i \in M, \forall \xi \in \Xi \tag{15}$$

$$I_t(\xi) = I_{t-1}(\xi) + \sum_{i=1}^{M} X_{it}(\xi) - D_t(\xi) - B_{t-1}(\xi) + B_t(\xi), \quad \forall t \in T, \forall \xi \in \Xi \tag{16}$$

$$\sum_{i=1}^{M} \sum_{t=1}^{T} X_{it}(\xi) \geq 0.8 \cdot \sum_{t=1}^{T} D_t(\xi), \quad \forall \xi \in \Xi \tag{17}$$

$$Z_i \in \{0, 1\}, \quad \forall i \in M \tag{18}$$

In the new model, the second stage now includes a new component, which is the production quantity per period for each selected supplier, letting first part only focus on selecting the suppliers.

In this work, our focus is to consider this problem as a two stage stochastic program. In the next two chapters, we will utilize two different approaches and in both the approaches we will treat this problem as a two stage stochastic program.

Chapter 4

Sample Average Approximation

In the previous chapter, we introduced the background and two models for inventory optimization in a compounding pharmaceutical company. In this chapter, we will present the Sample Average Approximation (SAA) algorithm, which can be utilized to solve this problem. Additionally, we will provide a detailed, step-by-step explanation of the algorithm and its application to the problem.

4.1 Introduction of the Sample Average Approximation

In two-stage stochastic programming, due to the presence of uncertainty, decision variables are divided into two distinct sets. The first-stage variables are determined prior to the revelation of the uncertain parameters that influence the second-stage decisions. Once the second stage parameters are decided, the recourse variables will be achieved. The whole objective is to minimize the first and second stage costs.

Two key challenges in solving the two-stage stochastic program are identified by Ahmed et al. (2002). One arises when accurately assessing the expected recourse costs: for a given first-stage decision and a realization of the random data, these costs are determined by solving an integer program. However, when dealing with a continuous distribution of uncertain parameters, estimating the expected recourse costs requires multi-dimensional integration. This process is often impractical and time-consuming due to its complexity.

Another significant challenge arises from the optimization of the expected recourse costs. Even if the expected recourse cost function could be straightforwardly evaluated or approximated, the overall stochastic programming problem requires optimizing this function over the set of first-stage decisions. The inherent difficulty lies in the fact that integer programming value functions are typically characterized by high non-convexity and discontinuities. This complexity leads to severe computational difficulties, as finding global optima in such landscapes is highly challenging.

4.2 Application of the Sample Average Approximation

The idea behind Sample Average Approximation (SAA) is to replace the expected value function with the sample average function. The sample average function is evaluated based on generating a Sample of a given size, say S, of the second stage random vector (demand in our case). The resulting problem is a good approximation to the original problem and much easier to solve. The SAA method converges to the true optimum. As the sample size increases, the accuracy of the solution improves. As this procedure is repeated, the SAA scheme not only generates high-quality solutions but also narrows the optimal gap(Ahmed, 2013).

The sample average approximation method involves generating a sample $\xi_1, \ldots, \xi_S$ consisting of S realizations of the random vector $\xi(\omega)$. This approach approximates the expected value function $E[Q(x, \xi(\omega))]$ by utilizing the sample average function(Contreras et al., 2011a).

As it calculates the recourse costs for each sample by solving the corresponding integer programming problem, and then averaging these costs to approximate the expected recourse cost, it efficiently address the complexity of the continue distribution of uncertain parameters. For the second challenge, the Sample Average Approximation (SAA) method transforms the original optimization problem into a formulation involving the sample average of the recourse costs. This reformulated problem can then be tackled using specialized algorithms designed for non-convex optimization.

$$\frac{1}{S}\sum_{s=1}^{S} Q(x, \xi_s).$$

We can denote $\hat{v}_S$ and $\hat{x}_S$ as the optimal value and an optimal solution, respectively, for the Sample Average Approximation (SAA) problem. Similarly, let v^* and x^* represent the optimal value and the optimal solution, respectively, in the real-world environment. Sample average approximation is employed to estimating the disparity of the two solutions. However, before we run the program, we need to solve two pivotal challenges. One arise from whether an increase in sample size will lead to an improvement in the accuracy of the problem's solution. Second, it is essential to investigate whether a balance between computational complexity and accuracy can be attained through a trade-off or by establishing a benchmark. (Ahmed et al., 2002) proved that in a linear optimization problem where both the first-stage and second-stage variables are continuous, the probability that the optimal solution of the Sample Average Approximation (SAA) problem will exactly match the optimal solution of the true problem converges to one exponentially as the sample size S increases. However, overly conservative to rely solely on increasing the sample size to enhance accuracy, given that the computational burden presents a practical challenge in real-world applications. (Kleywegt et al., 2002) proposed to use a efficient way to balance the sample size, solution quality and calculation burden by the formula below.

$$|S| \geq \frac{3\sigma^2_{\text{max}}}{(\epsilon - \delta)^2} \log \left(\frac{|B|}{\alpha} \right),$$

where $\epsilon > \delta$ and $\epsilon \in (0, 1)$. Here σ^2_{max} is the maximal variance of certain function differences Kleywegt et al. (2002). The method can only be applied to stochastic programs with the following characteristics Ahmed et al. (2002):

(1) **Finite First Stage:** The first stage of the program is finite. This implies that the number of initial decisions available is limited, although potentially very large.

(2) **Measurably Recourse Function:** For every x in X, the recourse function $Q(x, \cdot)$ is measurable, and the expected value $E[Q(x, \xi(\omega))]$ is bounded. This ensures that the second stage responses are well-defined and their expected costs are finite.

In our paper, it is imperative to conduct empirical testing to determine whether there is a positive correlation between the size of the sample and the reduction in the estimated gap. According to the design of our model, it is feasible to select a smaller sample size to estimate the range of the actual values associated with the real-world problem. Instead of calculating a large sample size for the Sample Average Approximation (SAA) problem, we generate a set of independent samples. Utilizing the solutions derived from these samples, we compute the lower and upper bounds for the optimal solution value. Based on the established principle, the following methodology will be implemented to compute the upper and lower boundaries. Subsequently, the magnitude of the discrepancy between these boundaries will be utilized to determine the optimal sample size.

4.3 SAA Algorithm in Model 1

Below are the step proposed by Contreras, Cordeau, and Laporte (2011b) to estimate the boundary of the original problem, The reason why we recommend to use in this paper, is because we have very similar background design and algorithm.

Step 1:

Construct a collection $K = \{S_1, \ldots, S_K\}$ of independent samples, each of size $|S|$, i.e., $\xi_i^1, \ldots, \xi_i^S$ for $i \in K$. For each sample S_i, solve the corresponding SAA problem subject to the objective function and the constraints of (19) to (23).

$$\text{Minimize} \quad \sum_{i=1}^{M} f_i \cdot Z_i + \rho \cdot \sum_{i=1}^{M}\sum_{t=1}^{T} X_{it} + \frac{1}{|S|}\sum_{s\in S_k}\left(\alpha \cdot \sum_{t=1}^{T} B_t(\xi) + \beta \cdot \sum_{t=1}^{T} I_t(\xi)\right) \tag{19}$$

$$\text{Subject to} \quad X_{it}(\xi) \le Z_i \cdot C_i, \quad \forall i \in M, \forall t \in T, \forall \xi \in \Xi \tag{20}$$

$$\sum_{t=1}^{T} X_{it}(\xi) \ge Z_i \cdot \text{MOQ}_i, \quad \forall i \in M, \forall \xi \in \Xi \tag{21}$$

$$I_t(\xi) = I_{t-1}(\xi) + \sum_{i=1}^{M} X_{it}(\xi) - D_t(\xi) - B_{t-1}(\xi) + B_t(\xi), \quad \forall t \in T, \forall \xi \in \Xi \tag{22}$$

$$\sum_{i=1}^{M}\sum_{t=1}^{T} X_{it}(\xi) \ge 0.8 \cdot \sum_{t=1}^{T} D_t(\xi), \quad \forall \xi \in \Xi \tag{23}$$

Step 2:

We will then compute the average and the variance of the previous SAA problem objective value.

$$\mu_K^S = \frac{1}{|K|}\sum_{i\in K} z_i^S,$$

$$\sigma^2_{\mu^S_K} = \frac{1}{(|K|-1)|K|} \sum_{i \in K} \left(z^S_i - \mu^S_K\right)^2 .$$

After we get the average value and the variance, we will treat the average value as the statistical lower boundary of the optimal problem. The variance of the $\sigma^2_{\mu^S_K}$ can treat as the variance of the estimator Norkin, Pflug, and Ruszczyński (1998).

Step 3: Select a feasible solution X^{S_i} of the original problem. In our approach, we will choose the most frequently appeared solution. By using this solution, it is possible to estimate the optimal value of the real problem with a larger sample size.

$$Z'_{S'}(\hat{Z}) = \sum_{i=1}^{M} f_i \cdot Z_i + \rho \cdot \sum_{i=1}^{M} \sum_{t=1}^{T} X_{it} + \frac{1}{|S'|} \sum_{s' \in S'_k} \left(\alpha \cdot \sum_{t=1}^{T} B_t + \beta \cdot \sum_{t=1}^{T} I_t \right)$$

In this step, we create a larger sample size S', starts from $\xi^i, \dots, \xi^{S'}$. Given the first stage two variables are stable, we can create a much bigger sample size (ten times in our case l S' to calculate the upper bound of the optimal solution by using aforementioned formula. Since it is an estimated upper bound, it is also important for us to measure the variance for the new sample size.

$$\sigma^2_{S'}(\hat{z}) = \frac{1}{(|S'|-1)|S'|} \times \sum_{s \in S'} \left(\sum_{i=1}^{M} f_i \cdot Z_i + \rho \cdot \sum_{i=1}^{M} \sum_{t=1}^{T} X_{it} + \frac{1}{|S|} \left(\alpha \cdot \sum_{t=1}^{T} B_t + \beta \cdot \sum_{t=1}^{T} I_t \right) - Z'(\hat{Z}) \right)^2$$

Step 4:

In the process of solving the problem, the lower and upper bounds are first established to evaluate the potential solutions of the Sample Average Approximation (SAA) method. The gap between these bounds is then utilized to estimate the effectiveness of the SAA program. Additionally, the obtained variance is calculated to further assess the dispersion of the solutions. This approach provides a better understanding of the performance, stability, and reliability of the SAA method in approximating the true optimal solution.

$$\mathrm{Gap}_{S,K}(\hat{z}) = \nu_S(\hat{z}) - \mu^S_K,$$

$$\sigma^2_{\text{Gap}} = \sigma^2_{\mu_K} + \sigma^2_S(\hat{z}).$$

In this chapter, we introduce the use of the Sample Average Approximation (SAA) method to estimate the lower and upper bounds for two-stage stochastic programming. Given that we are working with two models where the only difference lies in a single element in the first and second stages, their structural similarity allows us to apply the same procedure and SAA method to the second model as well. Consequently, to avoid redundancy, we will not provide all the detailed steps again, but rather implement the algorithm directly in the computational section.

Chapter 5

Lagrangean based Stochastic Decomposition Approach

In the previous chapter, we discussed the Monte Carlo sampling approach (SAA) to find the solution and estimate the lower and upper bounds. The accuracy of bounds depends upon the number of samples (K) and sample size (S). As we are solving an integer program, SAA approach is computationally burdensome. In this chapter, we discuss Lagrangian-based decomposition approach to obtain the lower and upper bounds of the problem.

5.1 Introduction of Lagrangean Decomposition

Lagrangian decomposition approach is prevalent in the integer programming literature. Many authors successfully tried this approach in the stochastic programming model too. The overall idea behind this approach is to relax the problem by dualizing a set of constraints with a set of multipliers (Lagrangian multipliers). This procedure reduces the problem complexity - frequently by decomposing the overall problem into multiple smaller problems. Solving multiple smaller problems provides a lower bound and any feasible solution provides an upper bound. The procedure is repeated with variety of multipliers in order to reduce the gap between the lower-bound and upper-bound. Given that our model includes integer variables in the first stage, according to (Carøe & Schultz, 1999), the stochastic integer model can be effectively solved using the dual decomposition method. This approach decomposes the second-stage problem into separate scenarios by relaxing the non-anticipativity constraints (NAC). Consequently, the problem is transformed into finding the optimal dual multipliers Oliveira (2021).

Various algorithms have been proposed for solving the dual problem: (Kelley, 1960) introduced the cutting plane method, while (Held, Wolfe, & Crowder, 1974) recommended the subgradient method. Some researchers, such as (Oliveira, 2021), have combined these two methods to the greatest extent possible, mitigating the limitations of each. Therefore, in our paper, we will adopt the method proposed by (Oliveira, 2021) to solve the problem. Additionally, in the computational section, we will compare the results of both the Sample Average Approximation (SAA) method and the Lagrangean Decomposition method.

5.2 Application on the Model 2

Before we proceed to build the Lagrangean decomposition algorithm, instead of solving both the two models we metioned before, only the second model is selected, since it allows us to more effectively evaluate the performance and robustness of the Lagrangean decomposition in more challenging scenarios, as the second stage contains more elements than model 1.

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{i=1}^{M} f_i \cdot Z_i + \left(\rho \cdot \sum_{i=1}^{M} \sum_{t=1}^{T} X_{it}(\xi) + \alpha \cdot \sum_{t=1}^{T} B_t(\xi) + \beta \cdot \sum_{t=1}^{T} I_t(\xi) \right) && (24) \\
\text{Subject to} \quad & X_{it}(\xi) \leq Z_i \cdot C_i, \quad \forall i \in M, \forall t \in T, \forall \xi \in \Xi && (25) \\
& \sum_{t=1}^{T} X_{it}(\xi) \geq Z_i \cdot \text{MOQ}_i, \quad \forall i \in M, \forall \xi \in \Xi && (26) \\
& I_t(\xi) = I_{t-1}(\xi) + \sum_{i=1}^{M} X_{it}(\xi) - D_t(\xi) - B_{t-1}(\xi) + B_t(\xi), \quad \forall t \in T, \forall \xi \in \Xi && (27) \\
& \sum_{i=1}^{M} \sum_{t=1}^{T} X_{it}(\xi) \geq 0.8 \cdot \sum_{t=1}^{T} D_t(\xi), \quad \forall \xi \in \Xi && (28) \\
& Z_i^1 = \cdots = Z_i^{|\Omega|} && (29)
\end{aligned}
$$

We add a new constraint (29), represents the non-anticipativity constraints, ensuring that first-stage decisions remain independent of any particular scenario and the second stage.

5.3 Application of the Algorithm

The core idea of dual decomposition is to create copies of the $Z^1, \ldots, Z^{|\Omega|}$ of the first stage (Carøe & Schultz, 1999). The first stage solution is required to satisfy all scenarios in the second stage. Constraint 23 represent the Non-anticipativity constraints (NAC) which can be separated into aggregated or disaggregated forms (Oliveira, 2021). The aggregated form uses a single constraint to express non-anticipativity, while the disaggregated form employs individual constraints to represent pairwise non-anticipativity. Choosing between these forms involves a trade-off. The aggregated form generally leads to weaker linear relaxations but requires fewer Lagrange multipliers, which may simplify the optimization process. On the other hand, the disaggregated form requires more multipliers, providing greater flexibility in finding optimal values but at the cost of increased computational (Oliveira, 2021).

Considering that our model focuses on pharmaceutical companies, which have higher demands for inventory management, reducing excess inventory is essential to prevent product expiration and waste. Therefore, we use the disaggregated representation to achieve more accurate results. Even the dis-aggregated method can be allocated to two manners, one method link the first scenario to all other scenarios, while another copy the previous scenarios variables. Those two methods are listed below.

$$\begin{cases} Z_i^{\xi} = Z_i^{\xi+1} \quad \forall \xi = 1, \ldots, |\Omega| - 1 \\ Z_i^{1} = Z_i^{\xi} \quad \forall \xi \in \Xi, \xi \geq 2 \end{cases}$$

Regardless of which of the manners used, the original problem is ultimately transformed into a solvable dual problem. The problem now change to the formula below.

$$\text{Minimize} \quad \rho \cdot \sum_{i=1}^{M}\sum_{t=1}^{T} X_{it}(\xi) + \alpha \cdot \sum_{t=1}^{T} B_t(\xi) + \beta \cdot \sum_{t=1}^{T} I_t(\xi) + \sum_{\xi} \lambda^{\xi} p^{\xi} \quad \forall \xi \in \Xi \tag{30}$$

$$\text{Subject to} \quad X_{it}(\xi) \leq Z_i \cdot C_i, \quad \forall i \in M, \forall t \in T, \forall \xi \in \Xi \tag{31}$$

$$\sum_{t=1}^{T} X_{it}(\xi) \geq Z_i \cdot \text{MOQ}_i, \quad \forall i \in M, \forall \xi \in \Xi \tag{32}$$

$$I_t(\xi) = I_{t-1}(\xi) + \sum_{i=1}^{M} X_{it}(\xi) - D_t(\xi) - B_{t-1}(\xi) + B_t(\xi), \quad \forall t \in T, \forall \xi \in \Xi \tag{33}$$

$$\sum_{i=1}^{M}\sum_{t=1}^{T} X_{it}(\xi) \geq 0.8 \cdot \sum_{t=1}^{T} D_t(\xi), \quad \forall \xi \in \Xi \tag{34}$$

The previously added constraint (29) represents the non-anticipativity constraints (NAC), which have now been transformed into $p^{\xi} = Z_i^{\xi} - Z_i^{\xi+1}$, in the sequential algorithm. Now the Lagrangean dual problem becomes finding λ such that $\nu_{LD} = \max_{\lambda} D(\lambda)$. Based on the fact that our original problem is a minimization problem, the optimal solution (dual optimal value) of the dual problem usually represents a lower bound for the original problem. By introducing Lagrange multipliers, the constraints of the original problem are relaxed, thereby constructing a dual problem. The objective of the dual problem is to maximize the dual function $D(\lambda)$ (Carøe & Schultz, 1999).

Since the problem at hand is a Stochastic Integer Programming (SIP) problem, there exists a duality gap, as duality theory suggests that the Lagrangian dual value ν_{LD} may be lower than the optimal solution of the primal problem (Guignard, 2003). Moreover, an essential characteristic of the Lagrangian dual problem is that it is a convex yet non-smooth optimization problem (Oliveira, 2021). This property allows the problem to be decomposed into smaller, independent sub problems, each corresponding to a specific scenario ξ. These sub problems can be solved separately, facilitating parallel computation, and the final solution is obtained by aggregating the results into the dual function $D(\lambda)$. The sub problem under each seario can be wrote as formula below:

$$D^{\xi}(\lambda) = \min \; \rho \cdot \sum_{i=1}^{M}\sum_{t=1}^{T} X_{it}(\xi) + \alpha \cdot \sum_{t=1}^{T} B_t(\xi) + \beta \cdot \sum_{t=1}^{T} I_t(\xi) + f^{\xi}(\lambda) x^{\xi}$$

Subject to Constraints (31)-(34)

where $D(\lambda) = \sum_{\xi} D^{\xi}(\lambda)$ and $f^{\xi}(\lambda)$ is given depending on the chosen formulation for the NAC. For the sequential case, we will use the method designed by Oliveira (2021).

Condition	$f^{\xi}(\lambda)$
$\xi = 1$	λ^1
$\xi = \|\Omega\|$	$-\lambda^{\|\Omega\|}$
otherwise	$\lambda^{\xi} - \lambda^{\xi-1}$

This approach decomposes the overall optimization problem into several independent subproblems, which simplifies the computational complexity and allows for parallel processing. By solving each subproblem separately, the solution can leverage the specific structure and data of each scenario ξ.

5.4 Algorithm of Solving Lagrangean Dual

The Lagrangian dual is a convenient computational approach; however, solving the full-scope problem relies on finding and improving the multipliers during each iteration. By optimizing these multipliers, the relaxation can be tightened, providing a closer approximation to the solution of the original full-space problem(Guignard, 2003). The algorithm stars by initial guess for a multipliers, normally by set a specific values (e.g,1), after each iteration, new relaxed solution will be achieved. The progress will stop until some criteria are satisfied. The algorithm progress flowchart are shown below:

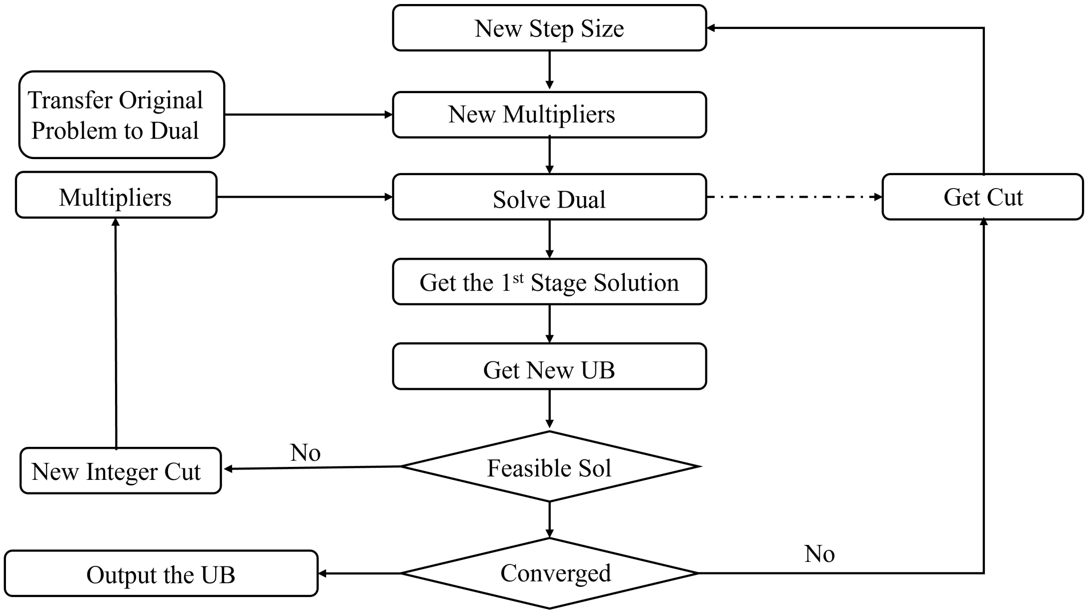

Figure 5.1: Lagrangean Decomposition Algorithm Flowchart

Initialization and Dual Problem Solving: The first iteration begins by generating an initial set of multipliers. These multipliers are then used to solve the dual problem, from which a cutting plane is derived. Following this initial step, the first-stage solution is obtained.

Estimation of the Upper Bound: In the second step, the first-stage solution is utilized to estimate an upper bound for the original problem.

Feasibility Assessment: The third step involves evaluating whether the estimated upper bound corresponds to a feasible solution. If the solution is feasible, the process advances to the fourth step. If the solution is not feasible, a new integer cut is generated, and new multipliers are created. The iteration then loops back to the first step to solve the dual problem again with the updated information.

Convergence Evaluation: In the fourth step, the algorithm assesses whether the solution has converged. If convergence is not achieved, a new cutting plane and an updated step size are generated, and the process iterates further. If convergence is achieved, the algorithm outputs the final upper bound as the solution.

To solve the Lagrangean dual problem, the most commonly used method is the subgradient method(Guignard & Kim, 1987). This method relies on subgradient information obtained after solving the Lagrangean relaxation problem. The subgradient information provides the direction of change of the function at the current point, which is used to guide the adjustment of the Lagrangean multipliers (i.e., the direction of improvement) and the magnitude of each adjustment (i.e., the step size). The step size determines the magnitude of the adjustment of the multipliers in each iteration, affecting the convergence speed and stability of the algorithm. This approach is often preferred for its computational simplicity. However, selecting effective strategies for defining and updating the subgradient step size is crucial. Despite its advantages, the subgradient method sometimes faces convergence issues. The other method in solving dual is called cutting-planes for approximating the Lagrangean dual function, which was propsed by Kelley (1960). Compared to the subgradient method, the cutting-plane method requires more iterations and is computationally intensive. However, the Lagrangian dual is more robust in avoiding convergence issues. The main challenge with the cutting-plane method lies in determining appropriate trust regions for the multipliers, as this approach is not particularly effective at generating optimal multipliers.

In this paper, two methods are combined to solve the Lagrangean dual problem and handle infeasible solutions: the subgradient Method and the cutting Plane Method. The subgradient method is used to provides approximated ascent directions. In each iteration, the algorithm calculates the subgradient information for each scenario and updates the Lagrangean multipliers based on this information. The Cutting Plane Method is employed to help to progressively approximate the solution to the original problem, reducing infeasibility and improving compliance with the problem's constraints. During each solve (Oliveira, 2021).

5.5 Upper Bound Procedure

One characteristic of the aforementioned algorithm is that the feasible solution can be used to estimate a valid upper bound for the full space problem. Oliveira (2021) mentioned that it is not always essential to compute the full space problem only when the full space problem is utilized for the purpose of decomposing into multiple scenarios. Besides, due to the first stage solution is applied to different scenarios, it is necessary for algorithm to identify and eliminate the infeasible solution. Therefore, the new constraint (35) is added in our model as an integer cut to exclude such solutions. By combining these methods, the algorithm ensures that infeasible solutions are effectively managed during the optimization process, progressively improving the feasibility and optimization results of the model.

$$\sum_{j \in Z_0} Z_j + \sum_{j \in Z_1} (1 - Z_j) \geq 1 \tag{35}$$

In this chapter, we introduced the use of the Lagrangean decomposition method for solving Model 2. By applying the dual decomposition technique, we can transform the original problem into a more manageable form. In the next chapter, we will test both the Lagrangian decomposition method and the Sample Average Approximation method using different datasets.

Chapter 6

Computation Experiment

In this section, we present our numerical experiments performed to test the practical applicability of both approaches in the context of our problem. We start with small instances to determine the appropriate sample size for the SAA approach. We need it to for setting the right trade-off between the calculation time and GAP. In addition, we have to make sure convergence of the algorithm too. To test the robustness and limitation of the Sample average approximation, we implemented the program in C++ with CPLEX as an optimization engine. Contreras et al. (2011a) proposed varying levels of nodes in their problem to test the practicality of the algorithm. Similarly, we categorized our suppliers into three distinct groups: small-size, medium-size, and large-size, each associated with different demand levels. For supplier sizes, we defined intervals as follows: [1, 5] for small-size, [10, 15] for medium-size, and [15, 25] for large-size suppliers. To simulate uncertain demand, we employed a normal distribution, adjusting the mean and standard deviation according to the coefficient of variation.

6.1 Convergence of the Algorithm Sample Average Approximation

In this section, to test the practical convergence of the sample average approximation (SAA), various sample sizes and numbers of samples are set (Contreras et al., 2011b). In the SAA section, we construct a collection of K independent samples, each with a sample size of S. Previous research (Ahmed et al., 2002) has indicated that as the sample size increases, the gap between the lower bound and upper bound decreases linearly. However, the increased sample size significantly impacts computation time. It is essential to get the trade-off between computational complexity and result accuracy. Given that the problem is set in a real-world environment, our focus is on identifying a configuration that reduces computation time while maintaining an acceptable gap between the upper and lower bounds.

We initiate the test for convergence of the SAA by set $|S| \in \{20, 50, 80, 100, 200, 500, 1000\}$ and $|K| \in \{10, 20, 50, 80, 100\}$ under the Model 1. For the numerical experimentation, we generated demand using Normal distribution with a mean of 3000 units with a standard deviation of 600. Other parameters for the experimentation are listed below:

Parameter	**Value**
Fixed Contract Cost for each Supplier	$C_F \sim U(1000, 3000)$
Max Capacity for each Supplier (per period)	$C_{max} \sim U(1000, 2000)$
MOQ (Minimum Order Quantity) (Over the planning horizon)	$Q_{min} \sim U(C_{max}, C_{max} \times 0.5 \times 12)$
Purchasing Cost	$= 1000/C_{\text{max}}$
Inventory holding Cost	4 \$ Per Unit
Backordering Cost	2 \$ Per Unit
Time Horizon	12 Months

Fixed Contract Cost: Each supplier incurs a fixed contract fee, $C_f \sim \mathcal{U}(1000, 3000)$, upon signing the agreement. This fee compensates suppliers for initiating production.

Max Capacity: The maximum capacity, $C_{\text{max}} \sim \mathcal{U}(1000, 2000)$, represents the highest number of units a supplier can produce in a month, chosen randomly between 1000 and 2000 units.

MOQ (Minimum Order Quantity): The MOQ, $Q_{\text{min}} \sim \mathcal{U}(C_{\text{max}}, C_{\text{max}} \times 0.5 \times 12)$, denotes the annual minimum procurement volume, ensuring supplier selected receive order higher than one month maximum capacity or not less than sum of six months' maximum production capacity.

Production Cost: The production cost decreases as maximum capacity increases, reflecting economies of scale. Lower capacities result in higher per-unit costs.

Inventory holding Cost: The inventory cost is set at 2 dollars per month for each stored unit.

BackOrdering Cost: The backordering cost is 4 dollars per unit per month, accounting for additional expenses to expedite logistics and fulfill delayed orders.

Time Horizon: We approached the problem from an annual perspective, thus setting the time horizon to 12 months.

These parameters model a dynamic supply chain with variable costs and capacities.

After running the experiment, according to figure 6.1 (the percentage gap between LB and UB), it is evident that as the sample size S increases, the percentage of the gap between the LB and UB decreases significantly for any given number of samples K. However, some fluctuations occur, such as when the sample size S equals 500, where the gap is higher compared to when S equals 200. This could be due to increased computational deviation when selecting a larger number of random demands. Additionally, we can observe that the spike at $S = 500$ is primarily due to the small number of samples $K = 10$. Consequently, we can conclude that increasing the number of samples K ensures more stable results, with the trend line showing a smooth decline. Conversely, when the number of samples K is small, even increasing the sample size S can cause some fluctuations. Therefore, it is necessary to use a larger number of samples K and a larger sample size S to achieve more stable and reliable results.

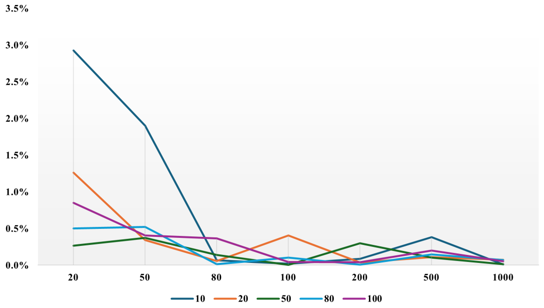

Figure 6.1: SAA Model 1 Percentage of Gap

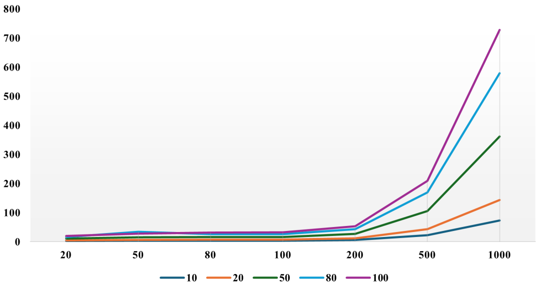

Figure 6.2: SAA Model 1 Calculation Time

Based on the computation time chart (figure 6.2), under a constant number of samples K, as the sample size S increases, both the computation time and complexity rise significantly. Additionally, for a given sample size S (referred to as scenarios), the computation time escalates with larger numbers of samples K. When the sample size S is set to 200, the computational complexity is similar to that observed with smaller sample sizes ranging from 20 to 100. Referring to our previous analysis of the gap percentage, the gap significantly narrows when the sample size is set to $S = 200$.

Considering the pronounced increase in computation time at $S = 500$ and the fluctuations in the gap at the same sample size, we have decided to give up using sample size more than $S = 500$. Additionally, the computational complexity at $S = 1000$ is prohibitively high. Given that the results at $S = 200$ are very similar to those at $S = 1000$ but require considerably less computation time, we have selected $S = 200$ as the optimal sample size. Figure 2 indicates that the line for $K = 80$ is more stable compared to others; therefore, to minimize deviations, we will employ the combination of $K = 80$ and $S = 200$ for testing under higher demand requirements and deviations, aligning with tests for medium and larger-sized supplier sets.

6.2 Experiment for the Small to Medium Size Problem for Model 1

In this section, we will use the combination of parameters $K = 80$ and $S = 200$ to test the SAA (Sample Average Approximation) method under different problem sizes. Our goal is to determine whether the SAA method has broad applicability.

Initially, we will conduct experiments with configurations involving 5 and 10 suppliers, alongside demand levels of 1000 and 3000 units, to represent small-scale problem scenarios. The coefficient of variation for demand randomness will be set at $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. Through this approach, our objective is to observe the performance of the Sample Average Approximation (SAA) method under these controlled conditions.

Next, we will increase the demand to 5000 units and test the SAA method with the same coefficient of variation for demand. This broader range will allow us to explore the stability and effectiveness of the SAA method when facing medium size problems with greater variability.

Through these two sets of experiments, we aim to comprehensively assess the performance of the SAA method under varying levels of complexity and provide more reliable results and insights.

Based on the provided data and analysis, the following observations can be made:

1: **Gap increase with the deviation coefficient**

As the deviation coefficient of mean demand increases, a significant expansion in the gap is observed. For instance, when the mean demand is 1000 units, increasing the deviation coefficient from 0.1 to 0.5 leads to an increase in the Gap from 129.9 to 866.7, with the corresponding percentage of the Gap rising from 0.23% to 1.09%. This trend indicates that increased deviation considerably

Mean of Demand	Standard Deviation of Demand	Supplier Number	Gap of the LB and UP	Percentage Gap of LB and UB	Calculation Time(s)	Supplier Selected
1000	5	0.1	129.9	0.23%	18.18	1
		0.2	247.2	0.40%	21.40	1
		0.3	397.0	0.59%	20.23	1
		0.4	779.7	1.06%	21.06	1
		0.5	866.7	1.09%	14.97	1
3000	10	0.1	416.5	0.28%	25.59	1, 3
		0.2	388.6	0.23%	33.08	1, 3
		0.3	1579.9	0.86%	33.00	1, 3
5000	15	0.1	123.3	0.04%	48.93	3, 15
		0.2	407.6	0.23%	47.96	3, 15
		0.3	294.5	0.01%	48.69	3, 15
		0.4	3572.7	1.08%	49.63	3, 15
		0.5	3751.6	1.04%	49.92	3, 15

Table 6.1: SAA Model 1 for Small and Medium Scale Problem

affects the complexity of the optimization problem, resulting in greater uncertainty between the upper bound and lower bound. This phenomenon is not only evident in small-scale problems, such as those with a mean demand of 1000 or 3000 units, but also in medium-scale problems with a mean demand of 5000 units.

2: **Computational Burden Increases with Demand Volume and Number of Suppliers**

The computation time is highly correlated with increases in both demand volume and the number of suppliers. The data reveals a clear upward trend in computation time as demand increases. Furthermore, as the number of suppliers increases, computation time grows exponentially. This indicates that in tackling larger-scale supply chain problems, the computational burden may influence the choice of the SAA method as a solution.

3: **Effectiveness of the SAA Method in High Deviation Problem**

The SAA method under our model demonstrates effectiveness when dealing with small-scale problems or when demand variability is relatively low. For instance, when the mean demand is 1000 units and the deviation coefficient does not exceed 0.3, the percentage of the gap is very small (a maximum of 0.59%). However, as uncertainty increases—particularly when the deviation coefficient rises to 0.4 or higher—the SAA method may struggle to provide the solution, especially in more complex scenarios. For example, when the mean demand is 3000 units, with 10 suppliers and a deviation coefficient of 0.4 or 0.5, the method fails to output the result due to the random parameters we generate for demand, capacity, and other elements, therefore in the Table 6.1 removed some results.

In summary, the data and analysis indicate that the SAA method encounters difficulties when addressing problems with randomly generated parameters, which may result in an inability to find feasible solutions and achieve tighter gaps.

6.3 Experiment for the Large Scale Problem for Model 1

Building on the insights gained from the analysis of small to medium-sized problems, we now extend our investigation to large-scale problems. This transition is essential to assess the scalability of the SAA method and to evaluate its performance under more complex and demanding scenarios, where the increased problem size introduces additional challenges and potential computational burdens.

We increase the demand from 5000 to 9000 and 15000, each corresponding to 18 suppliers and 25 suppliers, respectively. The deviation coefficient remains the same as in the small to medium-sized problem, and all other parameters and variables are consistent with those used previously.

Mean of Demand	Standard Deviation of Demand	Supplier Number	Gap of the LB and UP	Percentage Gap of LB and UB	Calculation Time(s)	Supplier Selected
9000	18	0.1	923.8	0.21%	60.97	3, 15, 16, 18
		0.2	944.4	0.19%	60.02	3, 15, 16, 18
		0.3	2126.3	0.39%	62.73	3, 15, 16, 18
15000	25	0.1	2914.2	0.39%	70.11	3, 15, 16, 19, 21, 25
		0.2	2820.0	0.34%	69.02	3, 15, 16, 19, 21, 25
		0.3	5920.5	0.65%	72.13	3, 15, 16, 19, 21, 25
		0.4	5124.6	0.51%	75.55	3, 15, 16, 19, 21, 25
		0.5	13328.6	1.23%	73.00	3, 15, 16, 19, 21, 25

Table 6.2: SAA Model 1 for Large Scale Problem

Based on our findings, it is evident that the SAA method continues to face the previously identified challenges when applied to large-scale problems. For example, when the mean of the demand equals 9000, the SAA method cannot output for result under deviation coefficient equals 0.4 and 0.5. Therefore, within the framework of Model1, we posit that the SAA method is effective for solving problems with small fluctuations, where the randomness remains within the bounds of feasible solutions. Given that the overall computation time remains manageable, the SAA method can be recommended as a viable approach for small-scale problems.

6.4 Sample Average Approximation Method Application for Model 2

As discussed in Chapter 3, to enhance the flexibility and dynamism of the problem, we introduce a slight modification to the model by shifting one element—the quantity produced by suppliers—to the second stage. To maintain consistency in the experiment, we continue using the fixed parameters from the previous chapter. This more complex model may yield outcomes that differ slightly from those obtained under the same parameter settings in the previous section. Additionally, we conduct experiments to determine the optimal sample size and scenarios, represented by K and S, for the new model. All other random parameters remain unchanged, including the demand, which varies from 1000 to 15000, the number of suppliers, and the deviation coefficient. The objective of this adjustment is to assess whether the changes in the first stage impact the results and the application of the sample average approximation method.

6.5 Best Combination for K and S for Model 2

In previous setups, the experiment specified $S = 200$ and $K = 80$ as benchmarks. Therefore, in this initial phase, we will constrain the value of K to a maximum of 80 and the value of S to a maximum of 200. We introduce a refined set of test sizes, $|S| \in \{20, 50, 80, 100, 200\}$ and $|K| \in \{20, 50, 80\}$, providing a more compact range compared to the previous section's SAA configurations.

The evaluation criteria include measuring computational time to assess the algorithms' complexity and analyzing the gap between the upper and lower boundaries to evaluate convergence. The uncertainty in demand for the test sets is controlled with a mean of 3000 and a deviation of 600. This structured approach is designed to rigorously test the adaptability and efficiency of the methods across different sample sizes and numbers of samples. The supplier number for all test are fixed, which equals to 6.

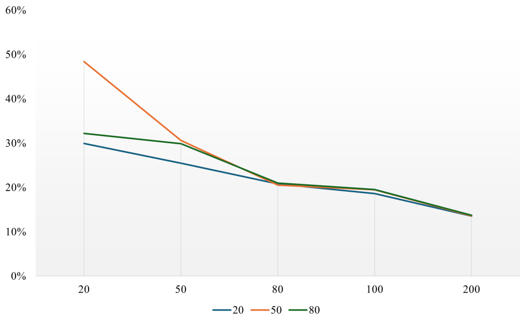

Figure 6.3: SAA Model 2 Percentage of Gap

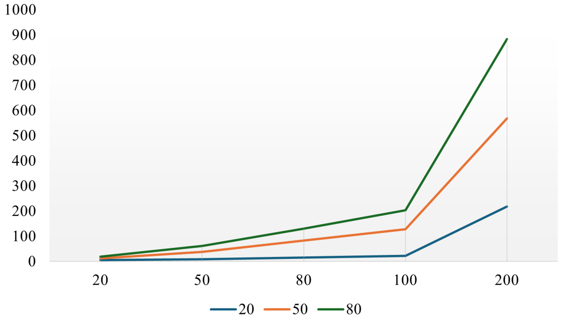

Figure 6.4: SAA Model 2 Computation Time

According to Figure 6.3, it is evident that as the sample size increases, the percentage of the gap decreases. However, the computational burden increases significantly based on Figure 6.4. Given that the difference in the gap between $S = 200$ and $S = 100$ is not significantly larger than when $S = 80$, yet the computational burden increases substantially, we will set $S = 80$. When comparing sample sizes of 50 and 80, we observed that there is no significant difference in the gap when the scenario count is set to 80. Consequently, this experiment adopts a configuration with a scenarios 80 and a number of sample of 50 for subsequent different-scale calculations.

6.6 Computational Results for Different Scales Under Model 2

Having determined the sample size and scenario count, we apply the SAA algorithm to Model 2 using demand that follows a normal distribution, ranging from 1000 to 15000, consistent with the parameters used in the previous model. The number of suppliers is adjusted accordingly, from 5 to 25, with the deviation coefficient varying from 0.1 to 0.5 to represent fluctuations. The problems are categorized based on the magnitude of demand: when demand is below 5000, it is classified as a small to medium-sized problem, while demand exceeding 5000 is considered a large-sized problem. The results for small to medium scale problem are shown below:

The results from Model 2 suggest that the Sample Average Approximation (SAA) method is not well-suited for problems where the second stage involves diverse elements and may not be the most appropriate approach for our model, as the complexity introduced in the second stage resulted in a substantial rise in the percentage gap. Furthermore, the computational burden increased

Mean of Demand	Standard Deviation of Demand	Supplier Number	Percentage Gap of LB and UB	Calculation Time(s)	Supplier Selected
1000	0.1	5	16%	29.19	3
	0.2		31%	29.18	3
	0.3		47%	30.14	3
	0.4		54%	38.54	3
	0.5		66%	45.27	3
3000	0.1	10	6%	81.82	1, 3
	0.2		25%	81.53	1, 3
	0.3		36%	81.60	1, 3
	0.4		52%	82.50	1, 3
	0.5		101%	88.37	1, 3
5000	0.1	15	12%	105.35	3, 12, 13, 14, 15
	0.2		27%	147.63	3, 11, 13, 14, 15
	0.3		43%	161.46	3, 10, 12, 13, 14
	0.4		76%	149.13	3, 5, 12, 13, 14, 15
	0.5		98%	159.65	3, 7, 10, 12, 13, 14, 15

Table 6.3: SAA Model 2 for Small to Medium Scale Problem

significantly, as reflected by the longer average calculation time compared to Model 1. However, the overall conclusions remain consistent with previous findings: an increase in deviation leads to greater computational burden and an expansion of the percentage gap.

In the context of large-scale problems, the inefficiency of the Sample Average Approximation method becomes evident. As we run the model for mean demand for both the 9000 and 15000, we found when the deviation coefficient increased to more than 0.6, the algorithm cannot output the result.

Mean of Demand	Standard Deviation of Demand	Supplier Number	Percentage Gap of LB and UB	Calculation Time(s)	Supplier Selected
9000	0.1	18	8%	180.05	1, 2, 3, 10, 11, 13, 14, 15, 18
	0.2		22%	180.34	1, 2, 3, 10, 11, 13, 14, 15, 18
	0.3		37%	185.39	1, 2, 3, 10, 11, 13, 14, 15, 18
	0.4		51%	188.20	1, 2, 3, 10, 11, 13, 14, 15, 18
	0.5		73%	195.30	1, 2, 3, 10, 11, 12, 14, 15, 16, 17
	0.6		105%	411.15	1, 3, 11, 13, 14, 15, 18, 20, 24
15000	0.1	25	5%	320.96	1, 2, 3, 5, 9, 10, 11, 13, 14, 15, 20
	0.2		19%	337.97	1, 2, 3, 5, 9, 10, 11, 13, 14, 15, 20
	0.3		18%	320.08	1, 2, 3, 5, 9, 10, 11, 13, 14, 15, 20
	0.4		43%	348.26	1, 2, 3, 5, 9, 10, 11, 13, 14, 15, 16, 19
	0.5		77%	385.65	1, 2, 3, 5, 7, 8, 10, 11, 12, 14, 16, 17, 19
	0.6		105%	399.56	1, 2, 3, 5, 7, 8, 10, 11, 12, 14, 16, 17, 19, 21, 22

Table 6.4: SAA Model 2 for Large Scale Problem

We analyzed the results for both models using the Sample Average Approximation (SAA) method and identified potential reasons for the increased gap percentage and infeasible solutions. The first reason is related to the algorithm itself, where, as mentioned in the introduction, the upper bound is estimated based on the initial solution obtained from the test. This approach can lead to issues with convergence. The second reason may lie in the inherent limitations of the algorithm we introduced, which, due to the randomly generated data, can lead to infeasible solutions, making it difficult to establish reasonable upper and lower bounds.

6.7 Convergence of Lagrangean Decomposition for Model 2

In light of the limitations encountered when implementing our algorithm with the SAA method in Model 1 and Model 2, we decide to apply Lagrangean Decomposition as an alternative approach for Model 2. To maintain the comparability of our results, we will continue to use the same fixed and random parameters. The problems will still be categorized as small, medium, and large scale, with means of demand were set as follows: Mean of the Demand $\in \{1000, 3000, 5000, 9000, 150000\}$. For the means of 1000, 3000, and 5000, the coefficient of deviation was varied within coefficient $\in \{0.1, 0.2, 0.3, 0.4, 0.5\}$.

However, distinct from the previous SAA application, we will expand the range of the deviation coefficient further, particularly for larger scale problems, such as mean of 9000 and 150000, the range was extended to coefficient $\in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$. This adjustment is intended to assess the robustness and applicability of Lagrangean decomposition across a broader spectrum of problem scenarios. Additionally, the number of suppliers was adjusted based on the demand scale and variability, with settings Supplier $\in \{5, 10, 15, 18, 25\}$.

6.8 Test for Small to Medium Scale Problem

We first conducted tests on small to medium-sized problems, with demand ranging from 1000 to 5000. According to the Table 6.5, in contrast to the results obtained from the Sample Average Approximation, the percentage gap in this case is within an acceptable range, with the largest gap between the upper and lower bounds being only around ten percent. Additionally, transitioning from small to medium-scale problems does not result in a significant increase in the gap. However, similar to the SAA model, the computational burden increases with the rise in both deviation and mean demand. It is worth noting that the Lagrangean decomposition method requires more computational time compared to the SAA method. Therefore, when dealing with small-scale problems rather than large-scale ones, the SAA method may be more favorable.

Mean of Demand	Standard Deviation of Demand	Supplier Number	Percentage Gap of LB and UB	Calculation Time(s)	Supplier Selected
1000	0.1	5	0.01%	227.15	3
	0.2		0.01%	229.19	3
	0.3		0.01%	227.64	3
	0.4		0.02%	280.66	3
	0.5		0.01%	325.56	3
3000	0.1	10	10.57%	357.08	1, 3
	0.2		10.14%	402.85	1, 3
	0.3		9.16%	454.38	1, 3
	0.4		8.20%	462.90	1, 3
	0.5		7.49%	500.08	1, 3
5000	0.1	15	0.01%	492.65	3, 12, 13, 14, 15
	0.2		5.40%	505.43	3, 11, 13, 14, 15
	0.3		5.25%	735.79	3, 10, 12, 13, 14
	0.4		4.83%	895.99	3, 5, 12, 13, 14, 15
	0.5		4.39%	1187.96	3, 7, 10, 12, 13, 14, 15

Table 6.5: Lagrangean Decomposition for Small and Medium Scale Problem

6.9 Test for Large Scale Problem

As the application of the Sample Average Approximation (SAA) method alone did not yield significant improvements in solving larger size problems, we moved to the Lagrangean Decomposition technique to solve larger-scale instances with demand scenarios specified at mean demands of $\{9000, 150000\}$, of which coefficient of variation within the range $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$. The performance of the Lagrangean Decomposition is shown below in Table 6.6.

It's evident that while the percentage gap increases with the mean and deviation, it remains within an acceptable range. In comparison to the SAA method applied under Model 2, the calculation time for Lagrangean decomposition is several times longer, indicating a greater demand on computational resources. However, Lagrangean decomposition has a distinct advantage over SAA: even when the deviation range is extended to 0.8 in large-scale problems, it consistently delivers results. This demonstrates its broader applicability and superior stability, especially in scenarios with high variability and stochastic uncertainty, making it a more robust solution than SAA.

Mean of Demand	Standard Deviation of Demand	Supplier Number	Percentage Gap of LB and UB	Calculation Time(s)	Supplier Selected
9000	0.1	18	10.95%	631.70	1, 2, 3, 10, 11, 13, 14, 15, 18
	0.2		10.46%	1305.63	1, 2, 3, 10, 11, 13, 14, 15, 18
	0.3		9.91%	1902.37	1, 2, 3, 10, 11, 13, 14, 15, 18
	0.4		9.30%	2094.84	1, 2, 3, 11, 12, 14, 15, 16, 17, 18
	0.5		8.57%	2544.03	1, 2, 3, 11, 12, 14, 15, 18, 20, 24
	0.6		5.26%	5492.52	1, 3, 11, 13, 14, 15, 18, 20, 24
	0.7		7.54%	3295.47	1, 3, 11, 14, 15, 18, 20, 24
	0.8		8.03%	4376.35	1, 3, 11, 14, 15, 18, 20, 24
15000	0.1	25	5.11%	2135.34	1, 2, 3, 5, 9, 10, 11, 13, 14, 15, 20
	0.2		13.20%	2978.19	1, 2, 3, 5, 9, 10, 11, 13, 14, 15, 20
	0.3		12.56%	3869.46	1, 2, 3, 5, 7, 8, 9, 10, 11, 13, 14, 15, 20
	0.4		11.64%	4724.29	1, 2, 3, 5, 9, 10, 11, 13, 14, 15, 18, 19
	0.5		9.82%	5213.25	1, 2, 3, 5, 9, 10, 13, 14, 16, 17, 18, 19
	0.6		10.14%	6042.25	1, 2, 3, 5, 7, 9, 10, 11, 13, 14, 16, 17, 21, 22, 25
	0.7		9.70%	7148.85	1, 2, 3, 5, 7, 8, 10, 11, 12, 14, 16, 17, 21, 22, 25
	0.8		9.28%	7086.60	1, 2, 3, 5, 7, 8, 10, 11, 12, 14, 16, 17, 21, 22, 25

Table 6.6: Lagrangean Decomposition for Large Scale Problem

6.10 Conclusion

6.10.1 Problem Overview and Research Conclusions

This research focuses on the inventory management challenges faced by compounding pharmaceutical companies. These companies often operate under resource constraints, whether in storage or procurement, and have received limited attention in existing research. However, the frequent drug shortages during the COVID-19 pandemic have led to increased attention on these businesses. In this thesis, we explored two different approaches to help these companies manage inventory pressure and reduce backorder risks in the face of uncertain demand.

Given the limited scope of prior research, we adapted methodologies from existing studies and applied two mainstream algorithm, utilizing Lagrangean decomposition and Sample Average Approximation (SAA) as our analytical tools. These models not only apply to compounding pharmaceutical companies but can also be extended to other small and medium-sized enterprises with similar operational characteristics.

The findings reveal that the Sample Average Approximation method is particularly suited for companies with a limited number of suppliers, stable product life cycles, and moderate demand fluctuations. SAA requires fewer computational resources, however, its output accuracy relies heavily on the application of a specifically designed algorithm for SAA, which is particularly crucial when generating simulated randomness. Additionally, the under performance of SAA in our tests may be due to inherent limitations of the model design, indicating that SAA could potentially yield better results when applied to multi-stage stochastic problems.

Conversely, the Lagrangean decomposition method is better suited for companies facing high demand variability, a diverse supplier base, and a need for more precise demand forecasting and management. Despite its higher computational resource demands, the Lagrangean decomposition method offers broader applicability and greater stability, particularly in environments characterized by significant stochastic uncertainty. Therefore, it is the preferred approach in such scenarios.

6.10.2 Limitations and Future Research Directions

Algorithm Design and Application: In developing the SAA algorithm, we closely followed methodologies from prior research. However, our study differs in context—while previous research primarily addressed Inventory Routing Problems (IRP), our focus is on multi-supplier selection and pharmaceutical production. This difference may have introduced certain biases, particularly in estimating the upper bound, where we used the first solution as a critical reference point. This approach could lead to infeasible solutions. Future research should focus on refining the criteria for selecting appropriate solutions to enhance upper bound estimation.

Scope of Products and Considerations: Given the unique nature of compounding pharmaceutical companies, our study was limited to a single product type rather than a multi-product context. Although we incorporated a multi-period model, other factors such as lead time and inventory turnover ratio were not fully considered. Future research should aim to integrate these factors for a more holistic analysis.

Application of Lagrangean Decomposition: We did not apply the Lagrangean decomposition method to Model 1, as Model 2's second stage, with its more complex supply chain and production dynamics, provided a more fitting context for demonstrating the advantages of this method. Future studies could explore modifying the SAA approach to accommodate a multi-stage stochastic framework and compare it with Lagrangean decomposition to more precisely delineate the appropriate use cases for each method. Additionally, while Lagrangean decomposition has proven effective and stable in optimizing large-scale problems, the associated computational resource demands remain a concern. Future research could investigate ways to improve the computational efficiency of Lagrangean decomposition, potentially through algorithmic enhancements or leveraging distributed computing techniques.

Multi-Stage Stochastic Programming: In this study, we utilized Two-Stage Stochastic Programming to address supplier selection and production quantity decisions. In our application of the SAA method, we encountered certain issues, which may stem from the fact that the problem we designed is a two-stage stochastic one

In contrast, Multi-Stage Stochastic Programming is better suited to address these challenges. By gradually introducing new decision variables and uncertainties at different stages—for example, selecting suppliers in the first stage, determining production quantities in the second stage, and incorporating sales volume and lead time constraints in the third stage—the multi-stage model can more accurately capture the dynamic nature of the supply chain.This approach may represent a valuable direction for future research.

Appendix A

My Appendix

A.1 Two-Stage Stochastic Model Code

This appendix provides the code developed for the Two-Stage Stochastic Model, which is the primary focus of this thesis. The code implements two key methods: the Sample Average Approximation (SAA) method and the Lagrangean Decomposition method. These methods were employed to solve the stochastic optimization problems discussed in the main body of this work.

The code can be accessed via the following link:

https://drive.google.com/file/d/1w_MbckJMGzQHQD1TN4phY0ugLNnp6wT7/view?usp=sharing

References

Abdelmaguid, T. F., Dessouky, M. M., & Ordóñez, F. (2009). Heuristic approaches for the inventory-routing problem with backlogging. *Computers & Industrial Engineering*, *56*(4), 1519–1534.

Adelman, D. (2004). A price-directed approach to stochastic inventory/routing. *Operations Research*, *52*(4), 499–514.

Ahmed, S. (2013). A scenario decomposition algorithm for 0–1 stochastic programs. *Operations Research Letters*, *41*(6), 565–569.

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

Alinaghian, M., & Zamani, M. (2019). A bi-objective fleet size and mix green inventory routing problem, model and solution method. *Soft Computing*, *23*, 1375–1391.

Aptel, O., & Pourjalali, H. (2001). Improving activities and decreasing costs of logistics in hospitals: a comparison of us and french hospitals. *The international journal of accounting*, *36*(1), 65–90.

Avella, P., Boccia, M., & Wolsey, L. A. (2018). Single-period cutting planes for inventory routing problems. *Transportation Science*, *52*(3), 497–508.

Azadeh, A., Elahi, S., Farahani, M. H., & Nasirian, B. (2017). A genetic algorithm-taguchi based approach to inventory routing problem of a single perishable product with transshipment. *Computers & Industrial Engineering*, *104*, 124–133.

Balkhi, B., Alshahrani, A., & Khan, A. (2022). Just-in-time approach in healthcare inventory management: Does it really work? *Saudi Pharmaceutical Journal*, *30*(12), 1830–1835.

Baum, N. (2006). Just in time'means more dimes in your pocket: Stocking only what your practice needs takes careful planning, but offers big savings. *The Bottom Line). Urology Times*, *34*(1), 28.

Campbell, A. M., & Savelsbergh, M. W. (2004). A decomposition approach for the inventory-routing problem. *Transportation science*, *38*(4), 488–502.

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

Charnes, A., & Cooper, W. W. (1963). Deterministic equivalents for optimizing and satisficing under chance constraints. *Operations research*, *11*(1), 18–39.

Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2012). The inventory-routing problem with transshipment. *Computers & Operations Research*, *39*(11), 2537–2548.

Contreras, I., Cordeau, J.-F., & Laporte, G. (2011a). Benders decomposition for large-scale uncapacitated hub location. *Operations research*, *59*(6), 1477–1490.

Contreras, I., Cordeau, J.-F., & Laporte, G. (2011b). Stochastic uncapacitated hub location. *European Journal of Operational Research*, *212*(3), 518–528.

Crainic, T. G., Fu, X., Gendreau, M., Rei, W., & Wallace, S. W. (2011). Progressive hedging-based metaheuristics for stochastic network design. *Networks*, *58*(2), 114–124.

Dantzig, G. B. (1955). Linear programming under uncertainty. *Management science*, *1*(3-4), 197–206.

Gianturco, S. L., Yoon, S., Yuen, M. V., & Mattingly, A. N. (2021). Outsourcing facilities and their place in the us drug supply chain. *Journal of the American Pharmacists Association*, *61*(1), e99–e102.

Guignard, M. (2003). Lagrangean relaxation. *Top*, *11*, 151–200.

Guignard, M., & Kim, S. (1987). Lagrangean decomposition for integer programming: theory and applications. *RAIRO-Operations Research*, *21*(4), 307–323.

Held, M., Wolfe, P., & Crowder, H. P. (1974). Validation of subgradient optimization. *Mathematical programming*, *6*, 62–88.

Jaillet, P., Bard, J. F., Huang, L., & Dror, M. (2002). Delivery cost approximations for inventory routing problems in a rolling horizon framework. *Transportation science*, *36*(3), 292–300.

Kelley, J. E., Jr. (1960). The cutting-plane method for solving convex programs. *Journal of the society for Industrial and Applied Mathematics*, *8*(4), 703–712.

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

Li, Z., Zhang, Y., & Zhang, G. (2020). Two-stage stochastic programming for the refined oil secondary distribution with uncertain demand and limited inventory capacity. *IEEE Access*, *8*, 119487–119500.

Linhares Rodrigues, A. (2019). Approximation algorithms for distributionally robust stochastic optimization.

Magableh, G. M. (2021). Supply chains and the covid-19 pandemic: A comprehensive framework. *European Management Review*, *18*(3), 363–382.

Mirzapour Al-e Hashem, S., & Rekik, Y. (2014). Multi-product multi-period inventory routing problem with a transshipment option: A green approach. *International Journal of Production Economics*, *157*, 80–88.

Moin, N. H. (2011). Optimization of multi periods inventory routing problem model with time varying demand. In *2011 ieee international conference on industrial engineering and engineering management* (pp. 190–194).

Nikzad, E., Bashiri, M., & Oliveira, F. (2019). Two-stage stochastic programming approach for the medical drug inventory routing problem under uncertainty. *Computers & Industrial Engineering*, *128*, 358–370.

Norkin, V. I., Pflug, G. C., & Ruszczyński, A. (1998). A branch and bound method for stochastic global optimization. *Mathematical programming*, *83*, 425–450.

Oliveira, F. C. P. (2021). *Petroleum supply chain management under uncertainty: Models and algorithms* (PhD thesis, Pontifícia Universidade Católica do Rio de Janeiro - PUC-Rio). Retrieved from https://www.maxwell.vrac.puc-rio.br/colecao.php?strSecao=resultado&nrSeq=55799@2 (Advisor: Silvio Hamacher)

Ramadhan, F., Imran, A., & Rizana, A. F. (2018). A record-to-record travel algorithm for multi-product and multi-period inventory routing problem. In *2018 5th international conference on industrial engineering and applications (iciea)* (pp. 274–278).

Solyalı, O., & Süral, H. (2017). A multi-phase heuristic for the production routing problem. *Computers & Operations Research*, *87*, 114–124.

Yu, Y., Chu, C., Chen, H., & Chu, F. (2012). Large scale stochastic inventory routing problems with split delivery and service level constraints. *Annals of Operations Research*, *197*, 135–158.