OPTIMIZATION MODELS FOR FRESH FRUIT PRODUCTION AND DISTRIBUTION

by

TRI-DUNG NGUYEN

Submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
November 2024

Dalhousie University is located in Mi'kma'ki,
the ancestral and unceded territory of the Mi'kmaq.
We are all Treaty people.

DEDICATION

I dedicate my doctoral dissertation to my parents, my wife, and my precious daughter, who have consistently provided me with unwavering encouragement and support. I am eternally grateful for your presence in my life.

Tri-Dung Nguyen

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Vietnamese dragon fruit leads the international market, accounting for over fifty percent of total production. 20%–15% of the dragon fruit output is consumed on the domestic market, while 80%–85% is exported. Asia, Europe, and the Americas constitute the primary consumer markets, wherein China contributes between 80% and 90% of the overall annual export value.

Nevertheless, the production and distribution of dragon fruit encounter various uncertainties and risks, including but not limited to weather conditions, pests and diseases, fluctuations in market prices, challenges related to logistics and infrastructure, competition, escalating input costs, sustainability concerns, and export regulations. This thesis presents research on mathematical models that aim to analyze and enhance quantitative modelling approaches utilized in the complex decision-making process of dragon fruit cultivation, harvesting, and distribution in Vietnam. By incorporating both deterministic and stochastic models, these models provide a methodical and data-centric strategy to tackle these obstacles. As a result, decision-makers are empowered to make well-informed decisions, which in turn enhance operational efficiency, minimize expenses, and improve the overall performance of dragon fruit supply chain management.

The decision-making process is structured into two stages: pre-planting, when costs and resources are considered deterministic; and post-harvest, when stochastic parameters become apparent. The decision-making process is divided into tactical and operational phases using the hierarchical planning approach, which is advantageous for cultivators, producers, distributors, and vendors.

The thesis has accomplished three of its objectives: conducting a comprehensive literature review on mathematical models that concern the production and distribution of fresh fruits, evaluating and developing a deterministic optimization model with certain data, and devising a stochastic optimization model to address the coupling and complex effects of uncertain factors.

LIST OF ABBREVIATIONS USED

ADB	Asian Development Bank
ARIMA	Autoregressive Integrated Moving Average
C&A	Collecting and Analyzing data
CCP	Chance-constrained programming
COVID-19	Coronavirus disease 2019
DC	Distribution Centers
DF	Dragon fruit
DMP	Distribution Matching Problem
DP	Dynamic Programming
DSS	Decision Support System
ECDF	Empirical Cumulative Distribution Function
ENSO	El Niño–Southern Oscillation
FADN	Farm Accountancy Data Network
FAO	Food and Agriculture Organization
FFSC	Fresh fruit supply chain
FL	Fuzzy Logic
FSC	Fruit Supply Chain
GA	Genetic algorithm
GLPK	GNU Linear Programming Kit
GMOs	Genetically Modified Organisms
GUSEK	GLPK Under Scite Extended Kit
HEU	Heuristics algorithms
KKT	Karush–Kuhn–Tucker
LP	Linear Programming
MILP	Mixed-integer linear programming
MIP	Mixed Integer Programming
ML	Machine Learning model
MONRE	Ministry of Natural Resources and Environment, Vietnam
MOSP	Multi-objective Stochastic Programming
MVT	Marginal Value of Time
ND	Network Designing
NLP	Nonlinear Programming
PMP	Positive Mathematical Planning
RO	Robust optimization
SAA	Sample average approximation
SDP	Stochastic dynamic programming
SFA	Stochastic Frontier Approach
SLP	Stochastic linear programming
SM	Simulation models
SMILP	Stochastic mixed-integer linear programming
SOFRI	Southern Horticultural Research Institute, Vietnam
SP	Stochastic Programming
STA	Statistical methods
TSSP	Two-stage stochastic programming

TH	Triple Helix model

STATEMENT

This thesis presents a collection of research exploring optimization models for fresh fruit producing and distributing, and it is comprised of a collection of research articles published or submitted for publication in peer-reviewed journals.

The earlier chapters (Chapters 1-4) lay the groundwork for the published articles included in the latter part of the thesis. These chapters establish the background, literature review, research statement and research objectives, and methodology used in the subsequent published works.

Chapters 5-7 (The Literature Review of Mathematical Programming Models for Fresh Fruit Supply Chain Management, The Deterministic Optimization Model for Fresh Fruit Supply Chains, and Stochastic Modelling Frameworks for Dragon Fruit Supply Chains In Vietnam Under Uncertain Factors) are based on published articles that offer unique contributions and further explore the research objectives established earlier.

Chapter 5 directly replicates the article titled “Mathematical programming models for fresh fruit supply chain optimization: a review of the literature and emerging trends” in *AgriEngineering*, 3(3), p. 519-541. I served as the first author on this publication.

Chapter 6, based on the article “Optimization Model for Fresh Fruit Supply Chains: Case-Study of Dragon Fruit in Vietnam” which appeared in *AgriEngineering*, 2(1), p. 1-26. In this work, I was the first author of this article.

Chapter 7, drawing on the published article titled “Stochastic Modelling Frameworks for Dragon Fruit Supply Chains in Vietnam under Uncertain Factors” in *Sustainability*, 16(6), p. 2423. I contributed as the first author on this publication.

These chapters, while published as separate articles, represent a unified research journey that progressively builds upon the knowledge gained in each preceding stage.

ACKNOWLEDGEMENTS

Prof. Dr. Tri Nguyen-Quang's enthusiastic guidance, sympathetic attitude, inexhaustible inspiration, and enlightened supervision enabled me to complete the research for this thesis. We will always remember and appreciate his diligent work ethic and commitment to upholding professional standards as a source of advice.

I express my sincere gratitude to the esteemed individual and my co-supervisor, Prof. Dr. Uday Venkatadri, for his expert guidance and unwavering support, which have been important in the successful completion of my study. I express profound gratitude to my committee members, Prof. Dr. Claver Diallo, and Prof. Dr. Michelle Adams, for their invaluable support, advice, dedication of time, and exceptional competence in assisting me with my project deliverables. Throughout the past years, my committee members have generously offered a substantial quantity of guidance and motivation.

Throughout my first four years of doctoral studies, the Ministry of Education and Training of Vietnam provided financial assistance, for which I am very thankful. The Research Grants of Prof. Dr. Tri Nguyen-Quang and Prof. Dr. Uday Venkatadri provided the financial support that enabled me to complete this work.

With acknowledging the invaluable contributions of my professors and friends—Prof. Dr. Ho Thanh Phong, Dr. Nguyen Van Chung, Ms. Bui Thi Kim Dung, Associate Prof. Dr. Phan Thi Mai Ha, Associate Prof. Dr. Do Ngoc Hien, Dr. Nguyen Vang Phuc Nguyen, Dr. Pham Duc Tai, Nguyen Huu Phuc, and Dr. Nguyen Vinh Du—my appreciation is insufficient for the accomplishment of this research work.

The affection and generosity of my lab companions and colleagues at Biosystems and Biofluid Modeling Lab, Dr. Muhammad Nadeem, Chinh Lieou, Dr. Van Hai Dang, Dr. Qurat Ul An Sabir, and Kayla McLellan, also contribute to my gratitude.

I am delighted to convey my profound appreciation and friendly regards to my cherished sister Nguyen Song Phuong Thao and her spouse, Dr. Nguyen Huynh Anh. Additionally, I wish to extend my gratitude to my cousin, Nguyen Vu Tung Chau. Ultimately, I owe everything to my devoted parents, Nguyen Duong Hung-Nguyen Thi Kim Oanh, my wife, the beloved Pham Binh Phuong Minh Ha, and my lovely daughter, Nguyen Tri Minh Khue,

who have persistently aspired to see me reach the pinnacle of success. They consistently raised their hands in devotion to me, and their assistance was indispensable in transforming this esteemed objective into more than just a fantasy.

CHAPTER 1 INTRODUCTION AND BACKGROUND

1.1 The Situation Of Production And Distribution Of Fresh Fruit In Vietnam

Fruits and vegetables are essential foods that cannot be absent from people's daily meals. The fruit and vegetable market in Vietnam has experienced significant swings in recent years. According to studies and analysis, the following information pertains to the production and consumption of fresh fruits.

The Vietnam General Statistics Office has reported that the present fruit tree area in the country is approximately 1.17 million hectares [1]. From 2010 to 2021, the average annual growth rate of the area is 3.1%. The southern region encompasses over 720 thousand hectares of fruit trees, or 62% of the country's total land. The Northern region encompasses over 445 thousand hectares, which represents 38% of the total land area of the country. The cultivation area of fruit trees has been steadily expanding each year, particularly for varieties that have seen increased demand for exportation in recent years, such as dragon fruit, durian, jackfruit, banana, mango, grapefruit, and others. In the period from 2017 to 2021, the average annual area of new planting for major fruit trees, especially in the southern region, is 62.4 thousand hectares.

The General Department of Vietnam Customs has reported that Vietnam's overall export value of fruits and vegetables experienced an average annual growth rate of 16.6% between 2011 and 2022. The diagram (Figure 1.1) provided illustrates the monetary worth of Vietnam's fruit exports from the year 2011 to 2023 [2].

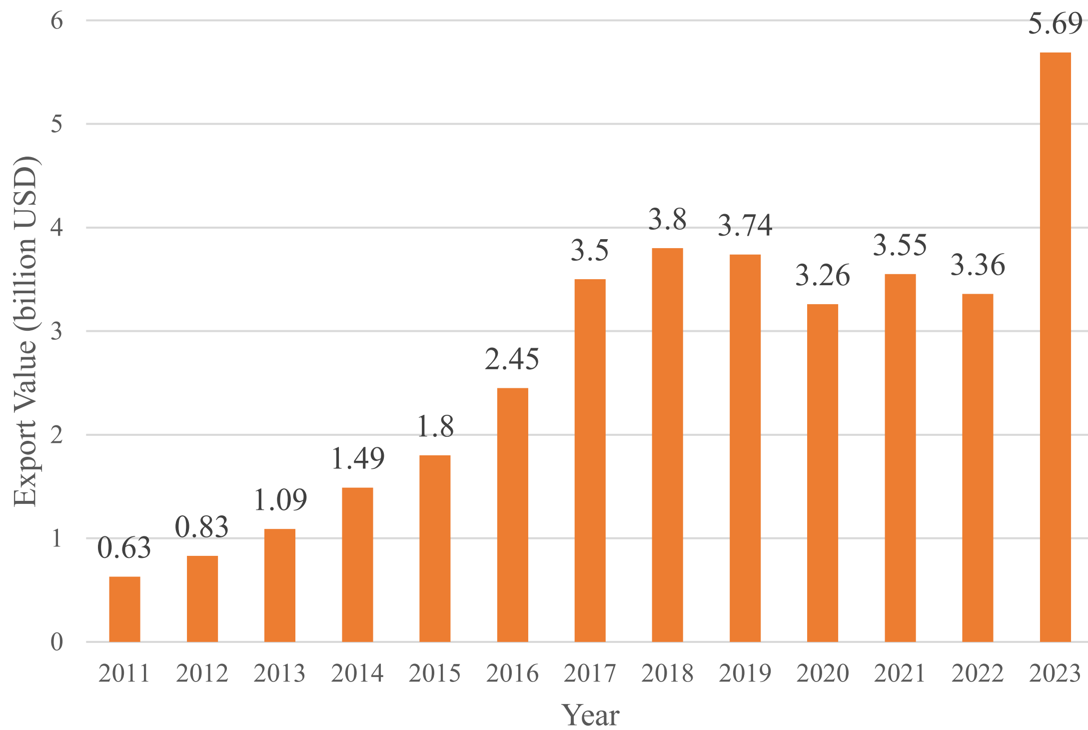

Figure 1.1 Vietnam's fruits and vegetables export value in the period 2010 – 2023

According to the direction of the Ministry of Agriculture and Rural Development, fruit exports are expected to reach a scale of over 5 billion USD by 2025 and are expected to increase to over 6.5 billion USD by 2030 [3], with an increasing rate of Compound annual growth rate (CAGR) is 5.42% during the forecast period (2024-2029). The markets such as China and South Korea are experiencing a growing demand for Vietnamese fruits and vegetables. The proliferation of free trade agreements is also bolstering the exportation of fruits and vegetables in the region. China accounts for about 50.0% of Vietnam's total fruit and vegetable export revenue. Nevertheless, endeavours to increase exports to lucrative markets such as the European Union, the United States, South Korea, and Japan have created favourable prospects for the country's fruits and vegetables. China, the United States, the Netherlands, Germany, Russia, Australia, Japan, Korea, Thailand, Taiwan, and Hong Kong [4].

In summary, Vietnam’s fruit production is robust, with a focus on fresh fruits, and the country continues to engage in international trade, particularly with China as a major partner.

1.2 CHALLENGES OF PRODUCTION AND DISTRIBUTION OF FRESH FRUIT IN VIETNAM

Vietnamese fruit farmers face numerous challenges in their efforts to cultivate and export fresh fruits. Some of these difficulties are:

- In recent years, the agriculture industry of Vietnam has seen specific difficulties as a result of the climate change that impacts the allocation of rainfall in the rainy season, leading to more frequent and intense huge storms and flooding in the upstream provinces of the Mekong Delta. Simultaneously, climate change extends the duration of the dry season, resulting in a decrease in the volume of water discharged from the primary channel of the Mekong River into the delta. This leads to a scarcity of freshwater and an escalation in saline levels [5]. Forecasts show that in certain provinces of the Mekong Delta, there will be a 25% rise in rainfall during the rainy season months and a 30-35% drop in rainfall during the dry season months by the year 2100. Consequently, the dry season will experience increased aridity while the rainy season will witness heightened precipitation [6]. Moreover, fluctuations in temperatures amplify the risk and impacts of diseases and pollution in the environment, both of which have a direct impact on agricultural output. Temperature rises have been observed at all sites in the Mekong Delta, with a projected increase of 1.8°C by the middle of the century and up to 3.7°C by the end of the 21st century [7]. The Mekong Delta, responsible for 60 to 65 percent of Vietnam's overall fruit export earnings, has faced unprecedented droughts and saltwater intrusion in the last century, particularly during the periods of 2015-2016 and 2019-2020, which were even more disastrous. The adverse circumstances in 2020 had a significant impact on around 25,000 hectares of fruit crops in the Mekong Delta. Ben Tre, Tien Giang, and Vinh Long provinces possess the largest fruit cultivation regions in the Mekong Delta. Nevertheless, they encounter productivity obstacles resulting from saline intrusion, which is a consequence of their coastal geographic positioning [8]. Experts predict that in 2024, there will be a severe drought that follows a 4-year cycle corresponding with the return of the El Niño [9].

- The infrastructure is still weak. Transportation is inadequate to cope with economic growth and connect with the outside world. The significant logistics expenses, amounting to 20-25% of the total product costs, reduce the competitiveness of Vietnamese agricultural products. Inadequate storage infrastructure results in post-harvest losses of up to 30%. Lack and insufficiency of warehouse systems, such as agricultural processing facilities and cold storage, limit efficient handling and storage [10] [11].
- The high incidence of refusal to import agricultural products from Vietnam persists due to non-compliance with international market standards, resulting in significant financial losses for manufacturers and exporters. The primary factors contributing to the rejection of agricultural products are bacterial contamination, improper labelling and hygiene control measures, pesticide residues, and the presence of additives. Furthermore, it is crucial to establish a comprehensive set of scientific criteria to assess the quality of agricultural goods during the process of exportation, along with explicit penalties for any breaches of these standards.
- The planning and managing production, distribution and pricing of fresh fruits are very complicated because their shelf-life is very short. The traditional practice of the trade is still influential in Vietnam. There are many intermediate nodes involved in the network, making the food supply chain longer and more complex than in other developed countries (Figure 1.2). The information in the value chain from growers to collectors/traders, wholesalers, retailers and supermarkets about the harvest, preliminary processing, packing, labelling, preserving, transportation is still poor, as is customer's awareness and usage of agricultural products.
- Farmers play the most important role in the food supply chain but most of them are small and unable to set the price. They sell their products at prices determined by the traders due to lack of market information and experience. Besides, although the development of agro-industrial sector rises rapidly, low paid labor is still used. Though the labor is cheap, there is a high turnaround. There is a workforce shortage in the beginning and the end of the season when demand is high due to planting and harvest respectively, offering opportunities for workers to change employers for better pay.

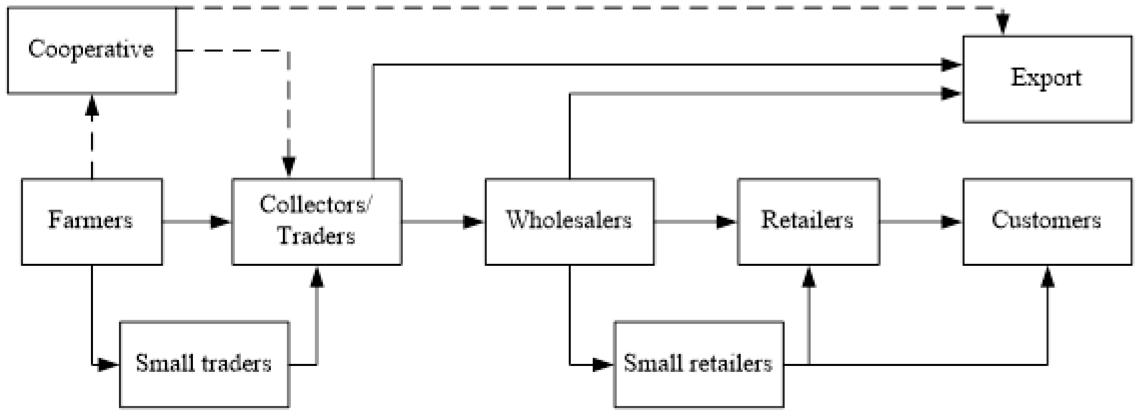

Figure 1.2 General fruit value chain in Vietnam

The value chain for fruits in Figure 1.2 shows two supply channels:

- The primary channel (straight arrows) involves farmers, traders, wholesalers, retailers, and customers. In this chain, the farmers who are so active, besides being owners of certain land growing fruit, they are also in charge of all work-phases from cultivation to consumption.
- The secondary channel (dashed arrows) shows groups of farmers in certain cooperatives. The cooperatives help them develop their produce (fruit), transportation, and markets.

1.3 DRAGON FRUIT

Dragon fruit plant (Hylocereus undatus), also known as the pitaya, strawberry pear, or night-blooming cereus, is a tropical fruiting vine native to southern Mexico and South America. The dragon fruit is a fast-growing and perennial plant which produces large white, fragrant, bell-shaped flowers that bloom at night, and fruits that are oblong with red or yellow skin and many small black seeds in the pulp. Although, the dragon fruit was introduced into Vietnam (Binh Thuan Province) by the French over a hundred years ago, it has really been developed into a line of merchandise from 1989-1990 [12] [13].

Today, dragon fruit is grown mainly in three provinces such as Binh Thuan, Tien Giang and Long An, and has grown and spread to 57 provinces throughout the country. According to Mr. Nguyen Quoc Manh (Department of Crop Production - Ministry of Agriculture and Rural Development), the whole country has nearly 55,000 hectares of cultivated land, and the output supplied to the market is 1,285 million tons of dragon fruit in 2022 (Table 1.1). From 2010 to 2020, the area of dragon fruit cultivation has a rapid growth rate of 15.1%/year and reached the highest level of 65,500 hectares in 2020. However, since then, the dragon fruit growing area in Vietnam decreased by more than 9,000 hectares compared to 2020 due to the impact of the Covid-19 epidemic and the China's zero-Covid policy [14].

Table 1.1 Planting area and output of the 3 largest dragon fruit provinces in 2022

Province	Planting Area (ha)	%	Quantity (tons)	%
Whole country	54,800	100	1,285,900	100
Binh Thuan	27,788	50.7	594,000	46.19
Tien Giang	9,000	16.42	307,500	23.91
Long An	8,300	15.15	262,800	20.44
Others	9,712	17.73	121,600	9.46

The first dragon fruit trees planted in Binh Thuan are Hylocerut undatus (red-skin and white flesh) belonging to the cactus family (Cactaceae) originated from Central and South America. Dragon fruit need a long process of photosynthesis for growing. The longer the sunlight lasts, the better the flowers are.

The dragon fruit variety with red skin and red flesh (Hylocereus costaricensis) was imported by the Southern Horticultural Research Institute (SOFRI)-Vietnam from Columbia in 1994 [15]. However, the red flesh dragon fruit is mostly exported to Chinese and Taiwanese markets because they like color red to decorate fruit trays in special events such as Lunar New Year [16].

A new cultivar of dragon fruit which is yellow-bur skin and white flesh dragon fruit (Hylocereus megalanthus) originating from Colombia was imported and experimentally planted by farmers in the 2010s. However, to meet the high domestic demand and the curiosity of customers, the fruit is imported from Malaysia and sold at the price that is up to 20 times higher than the red skin and white flesh ones [17].

Red-skin white-flesh   Red-skin red-flesh   Yellow-skin white-flesh

Figure 1.3 Species of Dragon fruit planted in Vietnam.

In such condition, dragon fruit trees blossom from April to September (favorable season) but most centralized from May to July when daytime is longer than nighttime (daytime lasts from 12.5 to 13 hours). From October to March, daytime is shorter, therefore farmers use electric power for lighting flowers. Dragon fruit was originally gathered annually during the summer season. Contemporary cultivators are now utilizing supplementary lighting to cultivate dragon fruit. This allows for year-round harvesting of the crop. This has facilitated the expansion of the fruit's market dominance and enabled the exportation of the crop to international markets and retailers.

Currently, although there are different varieties of dragon fruit grown in Vietnam, there are only two main commercial varieties of dragon fruit that are given priority for development in Vietnam: including red-flesh dragon fruit and white-flesh dragon fruit. Of these, white-fleshed dragon fruit is the main variety in Binh Thuan, accounting for 80% of the area. The red-fleshed dragon fruit variety is the mainstay in Long An, accounting for 97% of the area, and Tien Giang accounting for 71% of the area [18].

Vietnamese dragon fruit holds the largest market share in the international market, contributing more than 50% of global dragon fruit production [4]. The export value continuously increased from 57.15 million USD in 2010 to more than 100 million USD in 2011 and from 2017 to 2020, each year exceeding the one billion USD mark. In 2018 alone, the highest export value reached 1.27 billion USD. Figure 1.4.

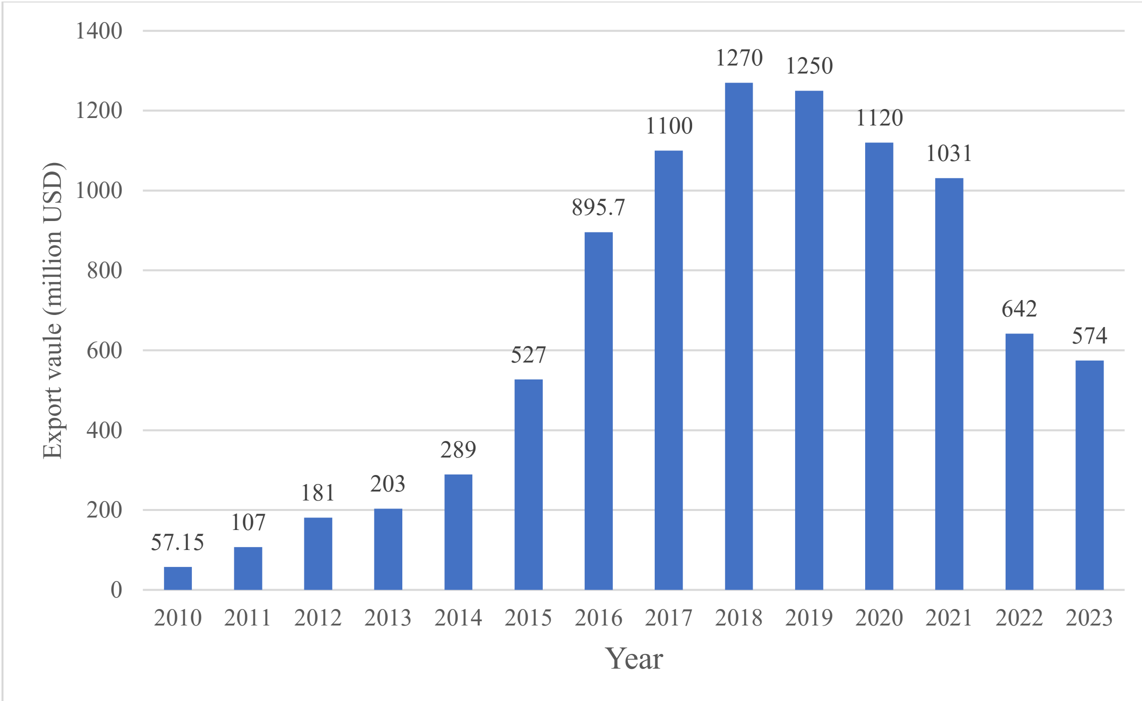

Figure 1.4 Vietnam's dragon fruit export value in the period 2010 – 2023

The domestic market consumes about 15-20% of dragon fruit output, the remaining 80-85% of output is exported. Vietnamese dragon fruit has been exported to more than 40 countries and territories, of which the main consumer market is Asia (China, Thailand, Indonesia, Hong Kong, Malaysia, Singapore...) accounting for 98% of volume and 91.1% in value; Europe (Netherlands, Spain, Germany, UK...) accounts for 0.9% in volume and 4% in value; Americas (United States, Canada, Chile...) account for 1.1% in volume and 4.9% in value [19]. China has always been the main export market for dragon fruits, accounting for 80-90% of the total annual export value.

1.4 Dragon Fruit Supply Chain And Challenges

Compared to other crops, dragon fruit production brings much greater economic benefits. However, as a sector in the agricultural industry, farmers, producers, and distributors are also facing uncertainties and risks such as:

- Weather Conditions: Dragon fruit cultivation is highly dependent on weather conditions. Extreme weather events such as typhoons, droughts, or heavy rains can adversely affect the quality and quantity of the fruit.

- Pests and Diseases: Dragon fruit plants are susceptible to various pests and diseases which can damage the crop if not properly managed. Controlling pests and diseases can be a challenge for farmers.
- Market Fluctuations: Fluctuations in market prices and demand can impact the profitability of dragon fruit production. Sudden changes in market conditions can lead to oversupply or undersupply situations.
- Logistics and Infrastructure: Ensuring efficient transportation and storage facilities is crucial for the distribution of dragon fruit. Inadequate infrastructure can result in spoilage of the fruit during transit.
- Competition: The dragon fruit market is competitive, both domestically and internationally. Vietnamese producers face competition from other countries including China, Isarel, India, Indonesia, Malaysia and Thailand in exporting their fruit. This competition can affect prices and market share.
- Quality Standards: Meeting the quality standards required for export markets can be a challenge for producers. Consistent quality control measures need to be in place to ensure that the fruit meets the standards set by importing countries.
- Increasing input costs (fertilizers, pesticides) affect competitiveness. Moreover, the planning and managing production, distribution and pricing are very complicated because the shelf-life of fruit is very short.
- Sustainability: Sustainable farming practices are increasingly important in the production of dragon fruit. Balancing productivity with environmental sustainability can be a challenge for farmers.
- Export Regulations: Adhering to export regulations and requirements of different countries can be complex. Producers need to stay informed about the regulations to ensure smooth exports.

1.5 Purpose Of The Thesis

Research mathematical models proposed in this thesis is focused on analyzing and improving comprehensive quantitative modelling approaches for the complex decision making of the harvesting and the distribution dragon fruit grown in Vietnam. The mathematical programming models are developed as a power tool for agricultural decision-making under uncertain factors.

The models offer a systematic and data-driven approach to address these challenges by optimizing various aspects of the fresh fruit supply chain. These models enable decision-makers to make informed choices, improve operational efficiency, reduce costs, and enhance overall performance in the supply chain management of fresh fruits.

The models that are built in this thesis can effectively address several challenges that the fresh fruit supply chain faces such as price fluctuations. They can assist in strategic decision-making by analyzing price trends and optimizing supply chain operations to adapt to market fluctuations.

CHAPTER 2 LITERATURE REVIEW

The fresh fruit supply chain is susceptible to uncertainties and risks that arising from multiple variables, such as climate change, water scarcity, and price volatility. The presence of these variables complicates the development of a comprehensive management strategy for the production, distribution, and pricing of fresh fruits. The globalization of the fruit and vegetable business has resulted in the creation of mathematical models and algorithms aimed at enhancing the efficiency of agri-food supply and fresh fruit supply chains. The application encompasses several functional domains, such as planting, optimizing harvests, managing logistics, and facilitating distribution. Researchers have proposed many mathematical models and techniques to improve the effectiveness of agri-food supply and the supply chains for fresh fruit. Most research focuses on the agri-food supply chain, with a specific emphasis on analyzing decision-making at several stages, such as production, storage, processing, transportation, routing, planning, and allocation.

Moreover, the utilization of mathematical models can aid decision-makers in systematically assessing and strategizing for many potential outcomes, particularly when managing a fresh fruit supply chain that is characterized by several uncertainties. Various modelling methodologies exist for analyzing the fresh fruit supply chain, such as deterministic and stochastic models. The deterministic optimization strategy is a commonly employed mathematical technique in the realm of fresh fruit supply chain management. There are three subcategories involved: linear programming (LP), dynamic programming (DP), and mixed integer programming (MIP). Caixeta-Filho [20] conducted a case study in Brazil to manage the scheduling of orange harvesting using a linear programming (LP) approach. The study examined the relationship between various chemical, biological, and logistical parameters and the quality of the harvested oranges. Hamer [21] utilized a decision support system in the form of an LP (linear programming) model to aid in the selection of suitable kinds of Brussel sprouts for planting. This approach involved discarding types that had unfavourable features. Widodo et al. [22] developed a supply chain model that integrates plant growth and loss processes into a mathematical framework. The objective of this model is to optimize the fulfillment of demands while minimizing losses in transit and storage.

Mixed Integer Programming (MIP) models are frequently employed to optimize fresh fruit supply chains. Examples include Maia et al.'s model [23] for scheduling routes for fruit and vegetable crops, Ferrer et al.'s model [24] for optimizing wine grape harvesting operations, Masini et al.'s tactical optimization model [25] for supply chains, Ahumada and Villalobos' operational MIP models [26, 27] for planning, and Jena and Poggi's MIP model [28] for sugar cane harvesting and processing. These models tackle different uncertainties in demand and yield, guaranteeing the development of optimal harvesting prototypes that satisfy market demand and optimize profits.

While the deterministic optimization approach is commonly used by researchers in fresh fruit supply chain studies, it is not capable of addressing high-risk concerns such as weather conditions or uncertainties in price fluctuation that might have an impact on decision-making in supply chain management. Stochastic programming and resilient programming can effectively handle the risks that arise in production and logistics planning within the agri-food industry. Robust optimization is a significant focus in the fresh fruit supply chain, enabling the management of uncertainties in real-world scenarios. Some examples of advanced techniques in agricultural planning and optimization are Bohle et al.'s resilient method [29] for wine grape harvesting, Munhoz and Morabito's model [30] for planning citrus production, and Darby-Dowman et al.'s two-stage stochastic programming model [31] for Brussels sprout planting and harvesting.

Stochastic models are essential for effectively handling uncertainty in the supply chain. The two-stage stochastic model developed by Kazaz [32] offers an optimal solution on a worldwide scale for production planning in the presence of unknown factors such as yield, pricing, and demands. Ahumada et al.'s planning model [33] for the production and distribution of perishable products also use stochastic methods to represent uncertainties, such as weather conditions and fluctuations in demand. The stochastic optimization approach developed by González-Araya et al. [34] focuses on optimizing resources and improving fruit quality in apple orchards, specifically for the goal of export.

Perishable production supply chains have been optimized using different mathematical models in several study fields. These models take into account variables such as cost and demand, ecological limitations, timing of planting and harvesting, aspects related to shipping and transportation, and styles of decision-making in operations. The stochastic frontier approach (SFA) [35] is a prevalent economic modelling technique that specifically addresses situations characterized by significant levels of randomness. The stochastic frontier strategy is employed to maximize farm production with the objective of enhancing profitability while minimizing any adverse effects on competitive markets. The machine repair models [36] have been adapted to enhance the efficiency of fruit harvesting and bin loading processes. This involves detecting and minimizing unproductive tasks, as well as enhancing labour and machinery schedules. The primary aspects to consider when planning and managing the fresh fruit supply chain are pricing and demand, environmental and decision-making approaches in operations. Future research should focus on integrating these techniques with meta-heuristics to address complex FFSC challenges on a broad scale.

In summary, mathematical models are crucial in facilitating contemporary data-driven decision-making approaches for managing the supply chain of fresh fruits. Future models should strive to integrate biological, environmental, and economic factors, including diseases, climate change, and price variations. Developing a database of FFSC models, organized based on local and regional criteria, would be highly beneficial for professionals and experts involved in modelling.

Based on the literature study, it can be concluded that two stage stochastic programming is an appropriate approach for managing food supply chains. This is because it effectively deals with the challenges of making optimal decisions in the face of uncertainty and helps managers mitigate future risks. However, there is a scarcity of models that address the agri-food supply chain in tropical Asian countries with high temperature and humid climates. The suggested model would take into account these aspects since they are crucial for preserving the quality and freshness of perishable fruits.

Furthermore, a limited number of models addressed the issue of operational planning. Growers, as the initial participants in the agri-food supply chain, play a crucial role in its success. Despite their slim profit margins, their activities such as timing for planting and harvesting, choice of planting varieties, fertilizer usage, water consumption, labour scheduling, and post-harvesting tasks must be carefully considered in order to ensure the effectiveness of the production and distribution plan. Put simply, effective operational planning can determine whether an operation is successful or unprofitable. Hence, the operational plan, which has been acknowledged as crucial for reducing costs and optimizing the quality of output, is an essential component of the suggested model.

The suggested research aims to analyze and enhance a comprehensive quantitative modelling technique for the intricate decision-making process involved in harvesting and distributing fresh fruit cultivated in Vietnam. Stochastic programming is a powerful tool used in agricultural decision-making when faced with uncertainty.

CHAPTER 3 RESEARCH STATEMENT AND RESEARCH OBJECTIVES

3.1 RESEARCH STATEMENT

The fresh produce supply chain encounters various obstacles, including protracted lead times, fluctuating demand and supply, and a narrow profit margin attributable to competitive forces. To enhance operational effectiveness and satisfy customer demands, supply chain managers must employ contemporary decision-making tools. Since the 1970s, mathematical models and simulation approaches have been proposed for fresh fruit supply chain planning. However, in light of various challenges including increased exports, quality and safety concerns, quantity, consistency, traceability, quarantine, packaging, labelling, uncertain market prices, intense competition from other exporting nations, and declining value growth, more robust tools are required.

Management of the fresh fruit supply chain requires a complex strategy for decision-making, particularly with regard to perennial plants whose optimum yield takes years to materialize. Modern data-driven decision-making strategies for medium-term and long-term production require the use of auxiliary tools. The most effective method for coping with FFSC containing numerous uncertainties, predicting scenario probabilities, and recommending trade-off-based decisions is mathematical modelling. The subject matter of this research thesis is the distribution and production of perennial fresh produce trees with rapid growth.

Based on the above contexts, my research question can be stated as "How the complexities will be explained and applied for the decision making in fresh fruit supply chain in Vietnam?

The subject matter of this research thesis is growth planning of perennial fresh produce trees with rapid growth. The thesis will present a comprehensive quantitative modelling approach for decision making in the fresh fruit supply in Vietnam, that has several complexities such as:

- A high number of transit nodes (cooperatives, traders, small traders, wholesalers, retailers, small retailers) can extend transportation times, increase handling costs, and potentially lead to spoilage.
- Fluctuating market prices during planting and harvesting can significantly impact farmer profits.
- Inefficient distribution networks can lead to delays, agricultural product damage, and higher costs for both growers and consumers.

By understanding the complexities of the fresh fruit supply chain and employing the right analytical tools and collaborative approaches, the farmers and managers in Vietnam can make informed decisions that lead to a more efficient, profitable, and sustainable fruit industry.

Dragon fruit is chosen to implement as a researching object for the following reasons:

- Although it is no longer the fruit with the highest export value in 2022 and 2023, dragon fruit is still a key export product with a value of hundreds of millions of dollars.
- Dragon fruit supply chain faces several uncertainties in yields, price, and demand, being highly sensitive to weather conditions and global uncertainty factors.
- The risk factors in the dragon fruit supply chain also depend on species.
- Although dragon fruit is used as the main target of the current study, our model can certainly be adapted to other fresh fruit production chains.

3.2 Research Objectives

To carry out the above research statement, three different research objectives are suggested as follows.

- Objective 1: Develop a literature review focusing on mathematical models to define uncertainties in the fresh fruit supply chain by pointing out the limitations and challenges, the need for mathematical models, the points strengths and weaknesses of existing models, and prospects for developing mathematical models for this research.

- Objective 2: Analyze and develop a deterministic optimization model which is a first step towards a comprehensive quantitative approach for decision making for fruit production and processing in Vietnam, with dragon fruit as the case study and main target for researching through some assumed price scenarios.
- Objective 3: Conceive and develop of a stochastic optimization model to deal with the complex and coupling effects of uncertain factors that could not be solved in Objective 2 for fruit production and processing in Vietnam, with dragon fruit as the main target.

CHAPTER 4 METHODOLOGY

To address uncertainties, decision making is typically guided by the two-stage decision making framework, a significant idea in stochastic programming as outlined in the literature review part of this thesis.

During the initial phase, known as the pre-planting stage, decisions on planting are made with the aim of maximizing anticipated revenue in the subsequent stage, referred to as the post-harvest stage. In the first stage, costs and resources are considered to be certain and predictable. However, in the second stage, decisions are made based on the results of uncertain factors such as crop output or market pricing. This framework has the capacity to handle both continuous and discrete distributions. It often requires creating scenarios to reflect the yield and market values of each product at the precise moment of shipment.

Furthermore, this study will employ a hierarchical planning technique, which divides the decision-making process into tactical and operational phases [37]. The hierarchical approach is suitable for several reasons, including the involvement of distinct decision makers, the presence of highly intricate problems, and the existence of decisions with varying time scales. Thus, in the hierarchical planning, choices are initially formulated at the tactical level and subsequently at the operational level.

The hierarchical planning offers the advantage of involving growers in decision-making processes about market and production. Coordinating tactical and operational decisions is advantageous for all stakeholders involved, including growers, producers, distributors, and vendors. The significance of this coordination has been observed through empirical evidence [33].

The hierarchical planning approach is utilized in a comprehensive model that is designed for both deterministic and stochastic models. This process involves two distinct phases:

- The initial phase involves strategic choices that are exclusively made at the beginning of the season, such as selecting the appropriate crop, determining the optimal quantity to cultivate, and deciding the appropriate timing for cultivation.

- During the second phase, farmers make adjustments to the decisions that were previously made in the first phase as the season proceeds. The farmers must determine the optimal quantity to harvest throughout each season and make strategic decisions regarding which customers to sell to, considering the prevailing market prices.

The proposed methodology is graphically summarized in Figure 4.1 below:

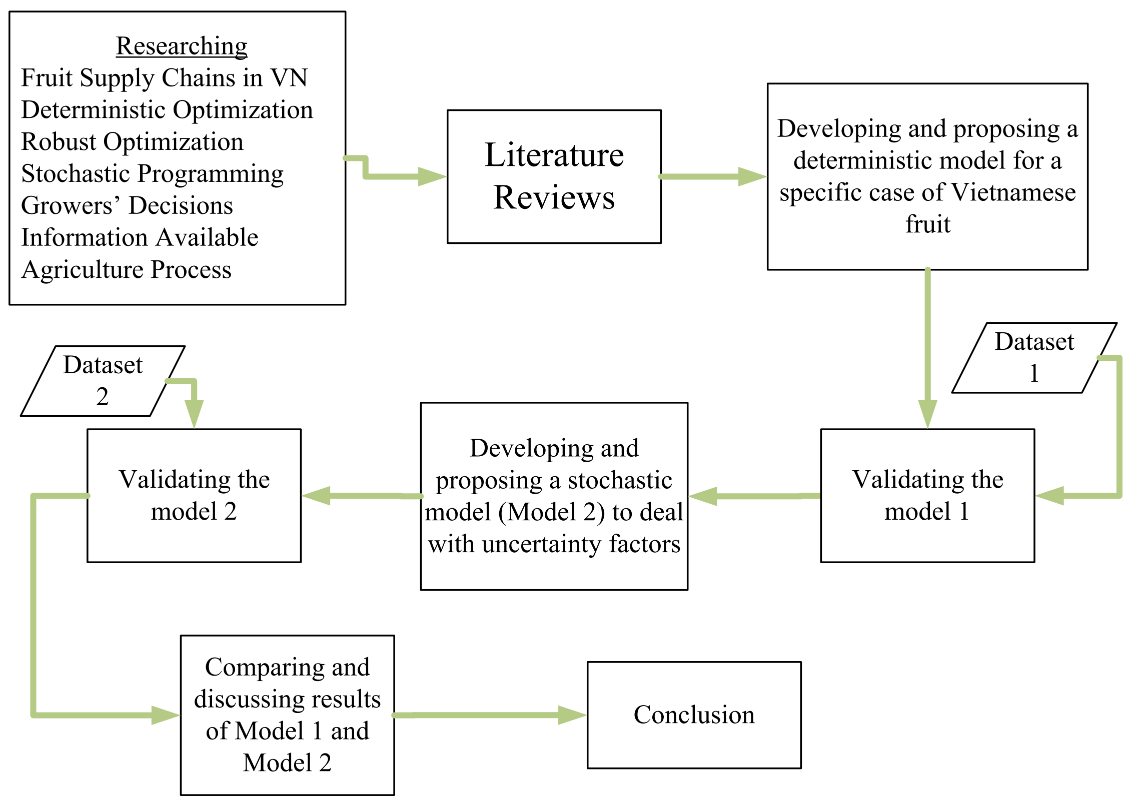

Figure 4.1 The methodology of the research

Based on Figure 4.1, the research work involves understanding the decision-making process of growers, the role of the fruit supply chain in Vietnam, and researching tools and methods for optimization problems. A deterministic model and stochastic approaches are proposed and applied sequentially to handle random variables such as market prices, weather conditions, biological conditions, demand, supply, and material flows in the food system.

To implement the hierarchical planning schematic, a deterministic optimization model is created specifically for the production of dragon fruit in Vietnam. The created deterministic model relies on anticipated and fixed parameter values to provide a comprehensive perspective for making strategic decisions in agricultural planning, including cultivation, harvesting, and final distribution to clients across various price scenarios. The deterministic model aims to optimize the planting and harvesting decisions to maximize the anticipated profit for the farmers. This profit is calculated as the difference between the projected total revenue from selling to traders, wholesale markets, and by-product suppliers, and the total costs associated with planting, truncating, by-product processing, penalty for demand shortfall, labor, lighting, and watering.

Ultimately, a recourse stochastic model is created to address the unpredictable parameters of agriculture by employing stochastic optimization and resilient methodologies, building upon the deterministic model. The model is responsible for managing medium- and long-term decisions related to crop planting, growth and harvesting, and distribution strategies for dragon fruit plants in Vietnam. A stochastic model is created to account for the random characteristics of dragon fruit chain production and distribution. The stochastic model is addressed using stochastic and robust strategies to handle the uncertain challenges related to fresh fruit production and delivery in Vietnam. The methods utilized in this study involve Scenario Tree Generation, Sample Average Approximation, and Chance-Constrained Programming. These approaches are created to understand and assess ambiguity to determine practical solutions to the problem. The inherent adaptability and capacity to handle increasing complexity of techniques [38] provide substantial benefits, making them powerful tools for solving challenges in stochastic programming. An analysis of the outcomes of both models is presented, highlighting the accomplishments and shortcomings, as well as proposing potential avenues for further research.

CHAPTER 5 THE LITERATURE REVIEW OF MATHEMATICAL PROGRAMMING MODELS FOR FRESH FRUIT SUPPLY CHAIN MANAGEMENT

This chapter is based on the published article "Mathematical programming models for fresh fruit supply chain optimization: a review of the literature and emerging trends" [39] in AgriEngineering, 3(3), p. 519-541. For more details, please refer to the electric copy that has been presented at https://www.mdpi.com/2624-7402/3/3/34.

5.1 Abstract

The fresh fruit agricultural and distribution sector is faced with risks and uncertainties from cli-mate change, water scarcity, land-use increase for industrial and urban development, consumer behavior, and price volatility. The planning framework for production and distribution is highly complex as a result. Mathematical models have been developed over the decades to deal with this complexity. With improvements in both processor speed and memory, these models are becoming increasingly sophisticated. This review focuses on the recent progress in mathematically based decision making to account for uncertainties in the fresh fruit supply chain. The models in the literature are mostly based on linear and mixed integer programming and involve variants such as stochastic programming and robust optimization. The functional areas of application include planting, harvest optimization, logistics and distribution. The perishability of the fresh fruit supply chain is an important issue as is the cycle time of cultivation and harvest.

5.2 Introduction

The consumption of fruits and vegetables is highly recommended for everyone on a daily basis, whether in developed, underdeveloped or developing countries, because of their content of essential vitamins, dietary fibers and minerals. Like other perishable products (fish, bread, packaged salads, and fresh meals), they indispensably require corresponding supply chains that can help deliver them from the original producers to the end consumer as fast as possible and in the best condition [40].

In recent years, along with the development of the supply chain management and logistics industry, the agriproduct supply chain in general, and the fruit chain in particular, have been recognized as a very important and strategic part of the economic development of many countries [41].

Compared to other stable and nonperishable crops, fruit production can bring greater economic benefits. However, like other sectors in agriculture, perishable fruit and vegetable production faces uncertainties and risks from society and the living environment, such as climate change, water scarcity, increase in land-use for industrial and urban development, consumer behavior and price volatility. It is very complex to build a management plan for the production, distribution and pricing of fresh fruits due to their short lifetime, seasonality in production, and the volatility of price and demand.

The fruit and vegetable industry has been globalized since the 1970s [42]. Nearly 700 million tons of fruits are produced worldwide each year, the most grown being bananas and apples, followed by grapes and oranges. Asia has always been considered the largest fruit bowl of the world due to large swaths of land in the tropics and subtropics and a high population, which the sector can count on for a readily available agricultural workforce [43]. The work force converted from cultivating various cereal crops (such as rice) so that fruit and vegetable production is also increasing [44, 45].

According to a report from the Food and Agriculture Organization (FAO) [46], in 2018 China took first place in the list of leading countries worldwide for fruit production, with an output of around 243.592 million tons/year, followed by India and Brazil, second and third place respectively. The yield of fruit from India is approximately 98.722 million tons/year, while for Brazil, about 40.047 million tons of fruit are produced yearly. Other large producers are listed in Table 5.1 below in order of their yearly quantity of production: The US, Turkey, Mexico, Indonesia, Spain, Iran and Italy [46].

Table 5.1 Top fruit producing countries in the world in 2018

Rank	Country	Fruits Produced (Million Tons)
1	China	243.592
2	India	98.722
3	Brazil	40.047
4	United States	26.015
5	Turkey	23.599
6	Mexico	22.768

7	Indonesia	20.436
8	Spain	19.332
9	Iran	18.898
10	Italy	18.009

A considerable number of mathematical models and algorithms have been proposed in the literature with the aim of improving the agri-food supply and fresh fruit supply chains; as well as several literature reviews related to the modelling approaches of agricultural supply chains have been done previously, and most of them just focus on the agri-food supply chain [47-50]. There are very few papers, such as those by Soto-Soto-Silva et al. [51] and Agarwal [52], which purely review the fresh fruit chain. For agriculture production models since the 1980s and earlier, a comprehensive literature review was done by Glen [47], while a revision for the crop production planning model was performed by Lowe and Preckel [48].

The review of Lowe and Preckel [48] focused on the agricultural facility allocation analysis to locate warehouses and processing plants. In addition, the complexity, challenges and uncertainties in strategic planning for production−distribution in the agricultural industry were considered in their proposed models.

Ahumada and Villalobos [50] referenced all the above-mentioned reviews but framed their review in the context of agriculture product supply chain planning. They basically took the same research approach that Lowe and Preckel [48] had adopted by considering only the production and distribution of crops. The authors covered crop production models developed for a few parties of the supply chain (including farmers and processing companies), but not for the macroeconomic models that could cover entire regions or countries.

Soto-Silva et al. [51], based on the work of Ahumada and Villalobos [50], has specifically reviewed operation research models related to the fresh fruit supply chain. The review emphasized on the rapid growth of science and technology in supply chain management to meet the challenges of increasing demand, high quality standards, and fierce competition in all aspects of production, processing and distribution for fruits and vegetables.

The review shared the same opinion as Ahumada and Villalobos [50] that although numerous papers on the agri-food supply chain have been published, most of them have just focused on a part of the supply chain. Factors such as decision-making levels, problems of production, storage, processing, transportation, routing, planning and allocation have attracted more attention from researchers. Both reviews ([47, 50]) showed that approaches such as linear programming and mixed integer programming are the most applicable methods in fresh fruit supply chains. Other models such as nonlinear programming, dynamic programming, stochastic programming or heuristic programming, although used less frequently, can also be relevant.

Our review focuses on the fresh fruit supply chain and looks at methodologies such as linear or stochastic programming models and their variants. We first look at various ideas leading to model formulations depending on market, environmental factors and agricultural fruit characteristics. This article is elaborated around the following points:

- Constraints and challenges in the fresh fruit supply chain
- The need of mathematical modelling in the fresh fruit supply chain
- Common concepts and dominant approaches
- Strengths and weaknesses of existing models
- Future perspectives

5.3 CONSTRAINTS AND CHALLENGES IN THE FRESH FRUIT SUPPLY CHAIN

5.3.1 Constraints and Challenges

The fresh fruit supply chain has a relatively long supply lead time, uncertain supply and demand, and a thin profit margin due to competition. These are the challenges that supply chain managers need to confront by improving efficiency and using modern decision-making tools [51]. In developed countries, where science and technology are better leveraged and crop productivity is high, production is not able to meet demand for several reasons; some of these are a shorter growing season in the North, demand for fruits and vegetables outside their seasons, and skilled labor shortages.

To meet the year-round demand for seasonal vegetables and fruits, most rich countries resort to a high level of imports in countries such as the US and Canada, up to 50% of fresh fruit is imported [53]. As fresh vegetables and fruits have better nutritional content and taste than preserved fruits/vegetables from past seasons, there is always sufficient demand for importing these internationally. In many developing countries, agriculture mostly follows traditional practices and uses manual labor. This brings unique challenges compared to developed countries which include difficulties in coordination between farmers, cooperatives, traders, wholesalers, distributors and retailers. Other challenges are:

- The traditional practice of trade is still dominant. With many intermediary stages as well as complex local rules, the food supply chain is longer and logistically more complex than in developed countries.
- Storage after harvesting and transportation is quite expensive due to a climate with high temperature and humidity.
- Although the growth of the formal agro-industrial sector has been rapid, the practice of using low paid labor is widespread. Though labor is cheap (and often unskilled), there is a high turnaround. Companies/farms must deal with workforce shortages during busy periods at the beginning and the end of the season when planting and harvesting take place, offering opportunities for workers to quickly change employers for better pay.
- Communication and the exchange of information between value chain partners in harvesting, preliminary processing, packing, labelling, preserving and transportation is often very poor, as is consumer awareness and the usage of agricultural products.
- Farmers are the most important factor in the food supply chain. However, most of them cannot set a good price for their products, due to these complex elements and their lack of market information and experience. The price for their products is often determined by traders, although cooperatives and fair trade have emerged through the last 50 years.

5.3.2 Influencing Elements

The influencing elements on the fresh fruit supply chains can be classified as follows (Figure 5.1):

- Functional areas: this category comprises production, harvest, storage and distribution.
- Purpose of the chain: this category includes the scope of the decisions made: such as harvest planning and optimization.
- Environmental factors: these include the planting environment with uncertainties and risks (countries with water shortage or natural calamities).
- Fruit characteristics, such as (1) highly perishable and (2) long shelf life.

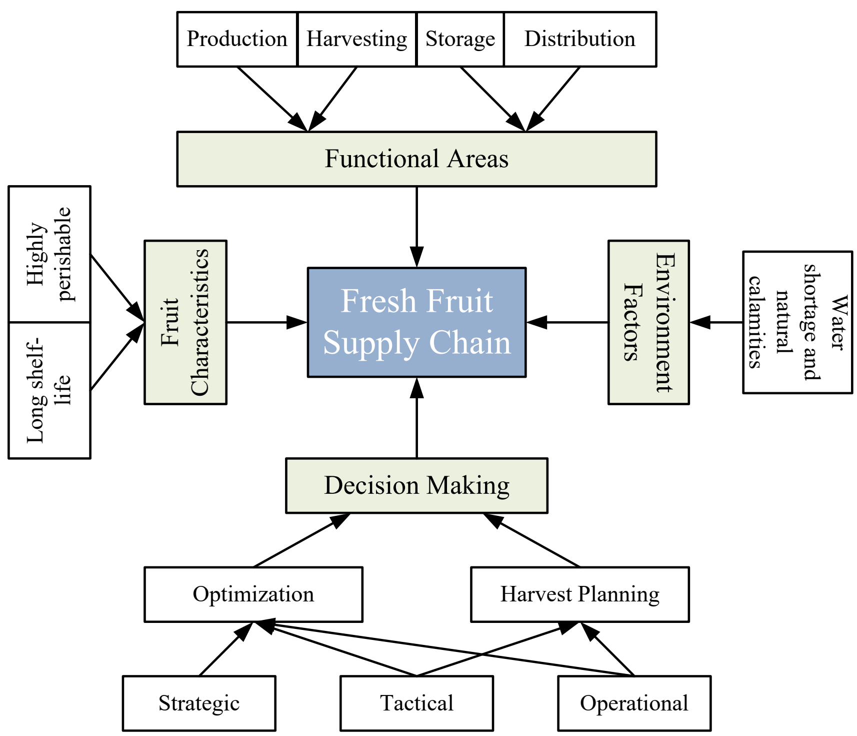

Figure 5.1 Constraints, challenges and influencing elements on the fresh fruit supply chain.

5.3.3 The Role of Mathematical Models

The decision-making strategy for fresh fruit supply chain management is a complex task, more difficult than with other supply chains [54]. This has remained a challenge for fresh fruit supply chain managers (FFSCs) over the past 40 years, especially in the context of increasing globalization and rapidly growing consumption. In the case of perennial plants that take years to grow and reach maximum yield, decisions taken today (changing the plantation portfolio, for example) have an impact on the future, based on the market demands. Therefore, we need tools to support modern data-driven decision-making strategies for medium-term and long-term production.

Table 5.2 Summary of mathematical models in references dealing with fresh fruit supply chain

Author	Model Approaches	Main Objective	Evaluation
Willis and Hanlon [55]	DP	Determine a plan to plant a variety of kinds of apples on the farm, using dynamic programming to optimize resources needed	Complex, mixed outcomes
Starbird [56]	DP	Determine a loading sequence for storage facilities at an apple packing plant by using a dynamic programming model	Complex, one outcome
Saedt et al. [57]	LP/MIP	Develop a plan to maximize revenue for a pot plant greenhouse with 2 models: one MIP for transition plans and one LP for future plans	Complex, mixed outcomes
Annevelink [58]	HEU	Use heuristic techniques to optimize the location of pot plants inside a greenhouse to minimize costs	Complex, mixed outcomes
Purcell et al. [59]	NLP	A quadratic programming model was developed for landscape land production to maximize returns for a given risk level	Complex, mixed outcomes
Van Berlo [60]	LP/MIP	Develop a tactical plan using a linear goal programming to minimize costs across the logistical chain	Complex, mixed outcomes
Hamer [21]	LP/MIP	Develop a plan using LP to support planting and harvesting decisions for Brussel sprouts with the objective of satisfying demand and maximizing profits	Complex, mixed outcomes
Maia et al.[23]	LP/MIP	Using MIP to make routing plans for fruit crops after harvesting with the objective of optimizing the capital investment under uncertainties	Simple, mixed outcomes
Miller et al. [61]	FL	A fuzzy program model was developed to minimize costs of production and harvesting at a tomato packing plant	Complex, mixed outcomes
Stokes et al. [62]	SDP	Using stochastic dynamic programming to make production and marketing decisions for a nursery producing ornamental plants with the objective of maximizing revenue	Complex, mixed outcomes
Broekmeulen [63]	HEU & SM	An assignment plan was proposed to improve the operations of a distribution center for fruits and vegetables using local search techniques	Complex, mixed outcomes
Leutscher et al. [64]	SM	Develop a simulation and regression metamodel to support making tactical and operational decisions for pot plant nurseries to increase profitability	Complex, one outcome
Darby-Dowman et al. [31]	SP	Propose a two-stage stochastic programming model to determine the optimal planting plans involving uncertain weather factors with the objective of maximizing revenue	Complex, mixed outcomes
Romero [65]	MOLP	A multiobjective model was built to find out an efficient cropping pattern by considering the risks for the farmers	Complex, mixed outcomes
Gigler et al. [66]	DP	Present a dynamic programming model as a methodology for optimization of agricultural product chains to deal with the appearance and quality of products	Complex, mixed outcomes
Hester and Cacho [67]	DP	Describing a dynamic model based on the complex biological and economic relationships of apple orchards to maximize the profit over a 15-year period.	Complex, mixed outcomes
Itoh et al. [68]	SP	A stochastic model was proposed to support crop planning dealing with uncertain factors with fuzziness and randomness to maximize revenue	Complex, mixed outcomes

Vitoriano [69]	LP/MIP	Presenting two mathematical models to compare, one with discrete time and another with continuous time, that support planning and scheduling tasks for crop production with given time horizon	Complex, mixed outcomes
Allen and Schuster [70]	NLP	A nonlinear model was developed to control risks of grape harvesting, to determine the optimal investing decision for harvesting and capital	Complex, mixed outcomes
Kazaz [32]	SP	A two-stage SP was applied for production planning under yield and demand uncertainty in olive industry, to maximize the satisfaction of customers and the profit	Complex, mixed outcomes
Rantala [71]	LP/MIP	MIP model was presented for solving all three problem levels of SCM: operational, tactical and strategic for a nursery company to minimize costs	Complex, mixed outcomes
Blanco et al. [72]	LP/MIP	MIP model was proposed to maximize the profit of a fruit packing plant by considering costs of raw material purchase, storage and labor.	Complex, mixed outcomes
Caixeta-Filho [20]	LP/MIP	Apply LP model to maximize the number of harvested oranges by considering quality factors.	Complex, mixed outcomes
Ortmann et al. [73]	LP/MIP	To optimize the export infrastructure, two models were presented: one for single product and other one for multiple products	Complex, mixed outcomes
Widodo et al. [22]	DP	Production, harvest and storage of fresh product were integrated in a periodical model that developed with growth and loss functions to maximize the demand	Complex, mixed outcomes
Ferrer et al. [24]	LP/MIP	To optimize costs of graph harvesting operations for wine production, a mixed LP model was used.	Complex, one outcome
Masini et al. [25]	HEU	A linear programming was presented to optimize a real fruit supply chain network to maximize profit.	Complex, one outcome
Bai et al. [74]	MOLP	To deal with fresh produce inventory control and shelf space allocation problem, an integration of four greedy heuristic methods was built to maximize revenue	Complex, one outcome
Cittadini et al. [75]	MOLP	A multiobjective linear programing model was proposed to maximize total profit and to optimize working force of an Argentinian cherry farm dealing with strategic and tactical plans.	Complex, mixed outcomes
Blackburn and Scudder [76]	SM	Developing a simulation model to optimize the value of marginal cost of a melon supply chain network.	Complex, one outcome
Van Der Vorst et al. [77]	SM	Introducing a new discrete event approach ALADIN to support decision making on redesigning a food supply chain with the objective of reducing costs and improving quality and sustainability.	Complex, mixed outcomes
Arnaout and Maatouk [78]	HEU	Dealing with scheduling problems of grape harvesting operations, some heuristic models were applied and compared with the objective of improving quality and saving costs.	Complex, mixed outcomes
Bohle et al. [29]	RO	Develop a robust model from an extension of stochastic model to deal with uncertain factors of operations of grape wine industry.	Complex, mixed outcomes
Morande and Maturana [79]	SM	An introduction of DSS based on simulation model for optimizing operations from harvesting to processing within the winery.	Complex, mixed outcomes
Arumugam et al. [80]	C&A	An analysis was made of supply chain of fresh fruit and vegetables to help Malaysian farmers in contract farming.	Simple, mixed outcomes
Verdouw et al. [81]	ND	Design a framework for the fruit supply chain network to support managers.	Simple, mixed outcomes
Ahumada and Villalobos [26]	LP/MIP	Propose an MIP model to maximize revenue from optimizing harvesting and distributing operations for bell pepper and tomatoes under uncertainty in short terms	Complex, mixed outcomes

Ahumada and Villalobos [27]	LP/MIP	An MIP was developed to deal with tactical operations of a vegetable supply chain for maximization of revenue.	Complex, mixed outcomes
Jang and Klein [82]	NLP	Develop a model to assist small farmers how to form and run a cooperative effectively, and then support them with the objective of optimizing quantity of milk production to contribute to a cooperative.	Complex, mixed outcomes
Jia and Huang [83]	C&A	A survey was conducted to study the relationship between cooperatives and buyers in China	Simple, one outcome
Rong et al. [84]	LP/MIP	Present an MIP model to optimize the plan of production and distribution of food supply chain with a target of increasing the food quality.	Complex, one outcome
Ahumada et al. [33]	SP	Propose a two-stage stochastic tactical model to deal with uncertainties of weather and demand in fresh vegetable industry and to support growing and distribution planning with the objectives of increasing revenue and decreasing losses.	Complex, mixed outcomes
Amorim et al. [85]	MOLP	A multiobjective model integrating operations of production and distribution of fresh products was built to minimize storing time.	Complex, mixed outcomes
Banaeian et al. [86]	C&A	To optimize energy for strawberry greenhouse and to increase strawberry yield, a nonparametric approach named data envelopment analysis was applied	Simple, one outcome
Perdana [87]	TH	The triple helix approach was applied to support all parties of the fresh fruit and vegetables supply chain in Indonesia to at all levels.	Simple, mixed outcomes
Yu et al. [88]	NLP	Use non-LP model approach to optimize inventory costs of both fast deteriorating and slow deteriorating products. The results of research showed that the total costs decreased significantly.	Complex, one outcome
Catalá et al. [89]	LP/MIP	Develop a mixed integer linear support for making strategic decisions in planting variety and density of pears and apples with the objective of maximizing the net present value.	Complex, one outcome
Bezat-Jarzębowska and Rembisz [35]	SP	A framework based on stochastic frontier approach was proposed to help the farmers with the objective of maximizing their profit.	Complex, one outcome
Jena and Poggi [28]	LP/MIP	Operational planning and tactical planning were integrated in a mixed integer linear model developed for optimizing sugar production with the objective of maximizing cane yield and profit.	Complex, mixed outcomes
Ampatzidis et al. [36]	ML	Apply a modified repair machine model to reduce harvesting costs by analyzing performance and scheduling workers and machines.	Complex, mixed outcomes
Lambert et al. [90]	FL	A modified Mamdani fuzzy model was used to increase production yield and fruit quality of Persian lime.	Complex, mixed outcomes
Munhoz and Morabito [30]	RO	Propose a robust optimization model developed from an LP model to optimize the midterm production plan of orange juice with the goal of minimizing costs	Complex, one outcome
Rocco and Morabito [91]	LP/MIP	A DSS based on mixed integer programing model was proposed to optimize operations scheduling and fuel logistics of steam production systems for tomato processing in Brazil.	Complex, mixed outcomes
Velychko [92]	LP/MIP	Develop a model that was integration of decision tree method and linear programming, to minimize operations costs and to maximize profit for every party of the fruit and vegetable cooperative.	Complex, mixed outcomes
González-Araya et al. [34]	LP/MIP	Present a tactical decision support system to optimize labor and resource scheduling during apple harvesting season with the objective of minimization of labor costs and maximization of quantity and quality apples to harvest.	Complex, mixed outcomes

Nadal-Roig and Plà-Aragonés [93]	LP/MIP	A prototype based on mixed integer program was proposed to support operational decision making for fruit logistic center to optimize transport planning with the objective of minimizing costs.	Complex, one outcome
Catalá et al. [94]	MOLP	Formulate a multiobjective integer linear programming to a pome supply chain including production, processing, distribution, and inventory stages with the objective of satisfying both two conflict goals as minimizing supply shortage and maximizing profit.	Complex, mixed outcomes
Rocco and Morabito [95]	LP/MIP	Form a prototype based on linear programming to support tactical planning in tomato processing industry in Brazil aiming to maximize profits.	Complex, one outcome
Grillo et al. [96]	LP/MIP	A multigoal programing model was developed to optimize a fruit supply chain in Spain with objectives of satisfying two conflict goals: maximizing total profit and minimizing shelf life of products.	Complex, mixed outcomes
Soto-Silva et al. [97]	MOLP	Three models covering actives such as purchasing, storing, and transporting apples of processing factories in Chile were developed to minimize costs.	Complex, one outcome
Cheraghalipour et al. [98]	HEU	The first proposed model was applied to minimized costs of the rice supply chain in Iran by implementing an integration of genetic algorithm and particle swarm optimization.	Complex, one outcome
Foong et al. [99]	LP/MIP	A mathematical model named input−output optimization model was developed to deal with palm planting and harvesting planning problems with objectives of maximizing yield but minimizing planting areas and gas emissions.	Complex, mixed outcomes
Gokarn and Kuthambalayan [100]	C&A	A study based on collecting and analyzing data was developed to evaluate uncertainties of the supply chain of fresh produce, and relationships among all outbound and inbound parties of supply chains of companies in India	Complex, one outcome
Ji et al. [101]	MIP/RO	To minimize the cost objective of a two-echelon inventory routing problem for perishable products, a robust optimization model was developed from an MIP model.	Complex, mixed outcomes
Varas et al. [102]	MOLP	Propose a multiobjective integer linear programming model to achieve conflicted goals that are maximization of harvesting quality and minimization of operation costs in Chilean wineries.	Complex, mixed outcomes
Alemany et al. [103]	MIP/FL	To deal with uncertainties in planting and harvesting fresh tomatoes, a fuzzy model developed from an MIP model was used to support decision makers with the objectives of maximizing income and minimizing costs.	Complex, mixed outcomes
Gómez-Lagos et al. [104]	MIP/HEU	An MIP model was proposed for tactical fruit harvest planning with the objective of minimizing the total cost by using greedy randomized adaptive search procedure metaheuristic method.	Complex, one outcome
Ktenioudaki et al. [105]	STA	To predict weight loss and to improve quality in blueberry processing, the boosted regression tree was implemented	Complex, one outcome
Lim et al. [106]	MIP/NLP	A harvesting and evacuation route optimization model was proposed to minimize travelling distance but maximize the quantity of palm harvested in Malaysia.	Complex, one outcome
Trivedi et al. [107]	MIP	Present a multistage integer linear program to optimize tactical transportation plans for apple supply chain in India with the goal of minimizing of costs and maximizing of demand.	Complex, mixed outcomes

C&A: collecting and analyzing data, DP: dynamic programming, FL: fuzzy logic, HEU: heuristics algorithms, LP: linear programming, MIP: mixed integer programming, ML: machine learning model, MOLP: multiobjective linear programming, ND: network designing, NLP: nonlinear programming, RO: robust optimization, SDP: stochastic dynamic programming, SFA: stochastic frontier approach, SM: simulation models, SP: stochastic programming, STA: statistical methods, TH: triple helix model.

With new international policies in the world related to fresh fruit importation/exportation [108] and consumer needs increasing year-by-year, mathematical modelling becomes indispensable. Mathematical models can assist decision makers to logically evaluate and plan for different possible outcomes. For example, when dealing with FFSC containing many uncertainties, mathematical modelling is the most effective tool to support decision makers by predicting scenario probabilities and suggesting decisions based on trade-offs [109].

Based on a literature review that covers over 70 articles linked to fresh fruit supply chains published within the last 40 years [41], there are two dominant categories of fruits that authors have focused on in their studies: (1) perennial crops, including apples [34, 42, 55, 56, 97, 104, 107], oranges [20], pears [25, 72, 89, 93], cherries [75], and grapes [24, 69]; and (2) annual crops including pineapple [77], strawberries [86], melon [76], tomato [26, 27, 33, 61, 85, 103] and blueberry [105].

Although many models and simulation approaches have been suggested in the last four decades, there is always a need for robust mathematical tools to support fresh fruit supply chain planning, due in part to the many challenges and constraints such as: (1) higher exports, quality and safety, quantity, consistency, traceability, quarantine, packaging and labeling; (2) fluctuating market prices; (3) hard competition with other exporting countries; (4) while exportation has been increasing in both volume and value, the growth in value has been declining [41].

5.4 Common Concepts and Dominant Approaches

Several modelling approaches for the fresh fruit supply chain were conceived based on various settings, constraints, challenges and influencing elements shown in Figure 5.2. Categorically, we can classify the models into four following groups: (1) models focusing on the functional areas (plantation, harvest, storage or distribution); (2) models based on the setting (decision-making scenarios for planning or optimization); (3) modelling environmental effects; and (4) modelling based on fruit characteristics.

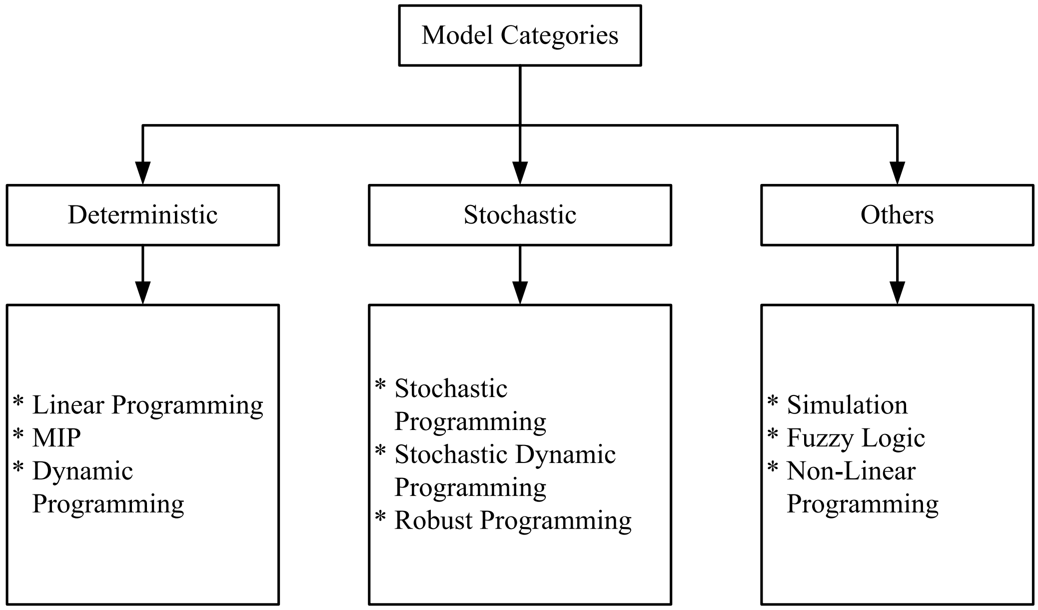

Figure 5.2 Model classification framework for fresh fruit supply chains.

From the modelling point of view, the classification of models falls into three main groups: (1) deterministic models and their variants; (2) stochastic models; and (3) special categories, which are neither deterministic nor stochastic (such as robust programming, which is a deterministic approach to model stochastic problems). Figure 5.2 shows our classification from the modelling viewpoint.

This chapter aims to identify, review, and classify research works dealing with the fresh fruit supply chain. We queried relevant research databases such as Web of Science, Google Scholar, Proquest with the following combination of keywords: mathematical model, fresh fruit, supply chain, agricultural supply chain and literature review. A total of 70 articles from 1976 to 2021 were collected and reviewed in this study. A brief description and the objectives to be achieved for each paper are presented in Table 2 along with a summary of the mathematical models and methods implemented by the authors. We also comment on the simplicity and outcomes of each model. From this summary, it is noticeable that linear programming (LP) and mixed integer programming (MIP) have been widely used. In addition, it is clear that the trend of using deterministic models is still popular. Although the supply chain of fresh fruit is influenced by many factors of uncertainty and risk, only a few authors have studied and applied stochastic models.

5.4.1 Deterministic Optimization Approach and Its Variants

It can be said that the deterministic optimization approach is very commonly used in FFSC research. There are three customary subcategories of models in this approach that researchers used: linear programming (LP), dynamic programming (DP) and mixed integer programming (MIP).

Caixeta-Filho [20] developed a case study using an LP approach for orange harvesting scheduling management in Brazil. The quality of orange fruit production is a competitive advantage of Brazil's citrus sector. Due to the fact that the harvesting time affects fruit quality, the author linked different chemical, biological, and logistical factors to the quality of harvested oranges. An actual dataset from 320 Brazilian farms producing over seven million boxes of oranges annually was collected to verify and validate the model. The contribution of the model is the application of fruit maturation information to orange harvest scheduling.

Hamer [21] used an LP for decision-making support for planting Brussel sprouts. Information on varieties, yield, customer demand and a scheduling method were all taken into account in the system. Varieties of fruits with bad characteristics such as shape, color or weak plants were eliminated in the model. The suggested linear programming model has supported producers in selecting varieties and then in providing a planting plan to meet market demand and maximize profits. Moreover, the model can be used to evaluate cropping in a tactical plan.

Dynamic programming models may also be found in the literature for FFSC planning. Based on the characterization of agricultural products by appearance and quality, Gigler et al. [66] suggested a model to deal with the optimization of agricultural chains to involve these two aspects. According to these authors, handling actions can influence appearance while the quality can be affected by processing, warehouse and transportation activities [66]. They optimize the "routes between components of the chain used to process products to minimize costs". Additionally, computer software has been developed based on dynamic programming methodology, in combination with the product quality development described as a function of the processing conditions.

Starbird [56] used dynamic programming for apple packing-plant operations. In this model, the author proposed that postharvest activities, especially storage delays, impact the quality of perishable agricultural products, which were quite similar to the model from Gigler et al. [66]. By determining the optimal order in which storage plant would be loaded, these delays were reduced. Besides, the optimal sequences depend much more on the deterioration rate of apple varieties than the storage facilities to which apples are transported to. However, this deterioration rate between two characteristics: hard and soft apples, was not reasonably determined. The optimal order did not additionally consider the difference between the capacity of warehouses and the number of apples which could be stored.

To reduce the losses of harvested fresh products, Widodo et al. [22] built a supply chain model by incorporating two important processes: plant growing and loss process into their mathematical formulation. They tried to apply the model for flowering−harvesting to maximizing the satisfied demands in every period of time and to minimizing the loss in the transportation and storage which could reach 60% of the total amount of harvested products. Their model also assumed that no on-hand inventory was carried through more than one period, and any requirement for harvesting fresh products should be satisfied through the earliest plant maturity as possible. The model suggested by Widodo et al. [22] could lead to an optimal harvesting prototype to maximize the satisfaction of demand levels. Some numerical examples were also presented by the authors to demonstrate the feasibility of the optimization algorithm.

The family of mixed-integer linear programming models seems the most commonly used and favored optimization techniques for the fresh fruit supply chain. For example, we can cite the model by Maia et al. [23] which addressed route scheduling for fruit and vegetable crops from the fields to markets. Based on various scenarios for alternative routes and a well-set condition of crops and markets, this model was used to optimize the capital invested in food preservation facilities.

The model by Ferrer et al. [24] optimized the scheduling of wine grape harvesting operations by considering both the quality of grapes and operational costs using the mixed integer programming approach. This model includes harvest planning, labor scheduling, and routing decisions by integrating quality loss in the objective function of the model. Their results showed the decisions which could be made at two levels, operational and tactical, to support the planning of the grape harvest at large vineyards.

Tactical level planning was also addressed by the MIP approach of Masini et al. [25], who developed the supply chain tactical optimization model in the fruit industry for a large fruit processing company in Argentina. The considerations in the model include demand from major markets, an estimate of fruit produced, capacity and availability of processing plants, a monthly storage plan for cold fruit, and final product delivery schedules. The model contains 18,000 continuous and discrete variables but does not incorporate uncertainties in demands and yield.

Ahumada and Villalobos [26] presented an operational model MIP type for planning to deal with harvesting and distribution in the fresh agricultural products industry. This model, which can generate short-term planning decisions for perishable agricultural products, was developed to maximize farmer revenues from production and distribution during harvesting. The model, through different approximations and simplified functions, can also deal with the availability of workforce, price fluctuation, influence of weather conditions, and biological properties and varieties of plants.

Continuing to develop their MIP approach, Ahumada and Villalobos [27] integrated the tactical planning approach into another framework which can assist the decision making for harvesting, packing and distribution to maximizing profit for growers of perishable products in Mexico. Decision making was based not only on traditional conditions (such as availability of workforce and price prediction), but also on factors such as spoilage of fresh products, and transportation and inventory costs. Authors also tried to tackle the complex planning issues of the supply chain management of perishable products. A loss function integrated into the objective formulation and storage constraints was used to consider perishability. With this model, growers could determine their planting plans, requirements of workforce and transportation throughout a crop season. In addition, the potential customers could be selected by the model via features such as the price they paid, the type of shipping they requested, conditions and quality of service they expected.

Jena and Poggi [28] applied an MIP model for sugar cane harvesting and processing for alcohol production. Their model was applied into two planning levels, tactical and operational, to maximize the total amount of sugar in the cane harvest. The tactical plan was designed for covering the entire seven-month harvesting season while the operational schedule was considered for the period from seven to thirty days.

Catalá et al. [89] presented an MIP model type for strategic planning optimization of pear and apple production. Their suggested model was used to assist a farm in optimizing its investment policy and in maximizing its current net value. The model has considered different financing scenarios to make dynamic decisions in a given period of time. Distinct constraints of their model place restrictions for risks, and integer decisions are linked to minimum planting area and the requirements for funding. Their results showed the optimal investment policy for the replacement of varieties under different scenarios, with and without external financing. In addition, a sensitivity analysis was applied for some of the mutually influencing parameters. The advantages of the model were that the modelling tool could be easily adapted to fresh fruits such as stone grape, citrus, etc., and in the explicit integration of financial considerations.

Three years later, Catalá et al. [94] extended their previous model [89]. A dual-objective model was proposed for tactical planning to maximize profit but minimize product supply shortage. A lexicographic method was developed to deal with the dual objectives. First, the inventory capacity, processing and shipping were optimized for profit, and second, the model minimized the shortage of supply.

To end this review of deterministic optimization models, Rocco and Morabito [95] developed a conceptual framework and mathematical model for production and logistics planning in the Brazilian tomato processing industry. The decision variables relate to allocating tomato areas, choosing tomato varieties, planning time for planting and harvest, routing transportation from fields to processing plants, scheduling to produce work-in-progress products (concentrated tomato pulps), final products to orders, managing inventories and shipping of these products to local warehouses.

Soto-Silva et al. [97] have developed an integrated model to optimize logistic activities for large apples in Chile. This integrated model was conceived to assist fresh produce purchase with the objective of minimizing the costs of purchasing and transportation between orchards and final destination location. This also included the varieties of apples of each producer, capacity of warehouses and the method of storage. According to these authors, their model could be applied to different fresh product processing companies using their own planting fruits in their processing [68]. However, the model did not consider uncertain factors such as weather conditions and truck fleets among producers, processing plants and warehouses.

Summary of section: Most papers dealing with FFSC in a deterministic context implement LP or MIP formulations to make tactical and/or operational decisions. Additionally, agricultural activities such as planting, harvesting and storing are covered more than the others. Besides, almost all authors only consider one kind of fruit as a case study to evaluate their model. Diverse decision-making levels and stages of the FFSC need to be considered more. Monoculture is known to be detrimental to soil health. Thus, future models should deal with polyculture farming and its SC implications.

5.4.2 Stochastic Programming Approach

Traditional deterministic models using linear programming or MIP are generally unable to deal with problems that involve uncertainties or give solutions with a high level of risk. This is particularly true in agriculture with several uncertain factors starting with weather conditions. Stochastic programming and robust programming (both extensions of linear programming) can address uncertainties in the parameters of linear or MIP optimization models for production and logistics planning in agri-food industries. Typical papers of stochastic or robust programming in the industry include Bohle et al. [29] and Munhoz and Morabito [30].

Robust optimization is also a major direction in the fresh fruit supply chain. In recent years, robust optimization has been used as a methodology that allowed uncertain factors to be confronted, especially if the probabilistic knowledge on the issues was incomplete.

A robust approach to optimize the harvesting scheduling for wine grapes with uncertain factors for which probabilistic knowledge may not be complete was developed by Bohle et al. [29]. The authors have considered scheduling problems for the wine grape harvesting subject to maximizing the actual yield being achieved during the harvest. The actual yield was considered as one of the uncertain factors when the schedule of wine grape harvesting was planned. To illustrate the robust optimization approach which could effectively deal with the uncertainties in practice, some alternative robust models were used to solve the actual problems for the wine industry. In addition to the real yield, the labor productivity could be also an uncertain element, reflecting various variable sources.

Unlike traditional formulations, according to the standard robust optimization theory, their proposed robust programming model controlled the variation of parameters simultaneously through different constraints instead of within the same constraint to achieve feasible solutions. An aggregate constraint was added with some goals as follows: (1) to reformulate the robust model; (2) to process simultaneously uncertain factors to the original constraints; (3) to avoid the worst case and (4) to have a higher chance of obtaining feasible solutions. Although the model developed by Bohle et al. [29] was heuristic, its solutions were feasible, and it kept the values of the objective function unchanged. The authors used the Monte Carlo simulation to evaluate the solution feasibility by running various scenarios.

Munhoz and Morabito [30] introduced an optimization approach for citrus production planning in Brazil. An aggregate planning model was developed for the production of frozen concentrated orange juice by using linear programming to make production, blending, and storage decisions based on orange maturation curves. The model has also integrated a robust optimization approach for some uncertain parameters.

A two-stage stochastic programming model for Brussels sprout planting and harvesting with the utility functions was proposed by Darby-Dowman et al. [31] to minimize risks incurred by farmers. In the first stage, a planting plan for all scenarios is created and then a harvesting schedule for each scenario is developed in the second stage. The stochastic optimization model takes into account uncertainties related to the biological nature of crop production, weather and environment conditions, as well as changing demands that could impact prices. Therefore, vegetable producers can develop planting and harvesting plans that are more reliable and more robust than in a deterministic model, even though they may be less profitable on average, based on risk parametrization.

Kazaz [32] used a two-stage stochastic model for production planning in the olive oil industry under uncertainty in yield, prices and demands. Starting with studies on random yield, the author defined sale prices and purchasing costs as exogenous inputs and inversely proportional to yield. The model assumes a yield-dependent price and purchasing cost based on random yield and demand. Kazaz's model [32] had four main contributions: (1) the objective function was concave if the planting areas were leased, there-fore a globally optimal solution was provided by first-order conditions; (2) the model has illustrated how changing yield could affect the total production of olive oil; (3) the optimal farm space leased was proven to be decreasing if a second (and reliable) source of supply appeared; and finally (4) unlike in traditional maximizing yield modelling approaches, the model showed the yield would increase even if the optimal farm space leased was not expanded and even when there was a second supply source.

Distinct from the deterministic approach, the planning model by Ahumada et al. [33] for production and distribution at the tactical level of perishable products relies on the stochastic approach to model uncertainties such as weather and variability of demand. Another feature of the model is the ability to choose different risk preferences for greater robustness.

A stochastic optimization approach for harvest planning in apple orchards was developed by González-Araya et al. [34]. This model considers resource optimization as well as higher fruit quality for export purposes. The mathematical model constraints include satisfying orders generated by the fruit packing plants, plant operations with the right capacity, production at the fruit orchards, and the right harvest time for each variety of apples. A real dataset collected from three orchards in Chile during two harvest seasons were used to explain the model.

Summary of section: This section reviewed the articles using stochastic methods to deal with uncertain factors in the FFSC. The L-shaped method is considered an effective tool to solve the stochastic problem. In addition, most of the authors believed that the two-stage stochastic model was a good choice for making tactical and operational decisions. Hence, two-stage stochastic models will still be used to deal with risks and uncertainties in the

FFSC. However, new developments in robustness should be considered and applied to support decision making under uncertainty.

5.4.3 Special Category Models

In many fresh fruit supply chains, uncertain elements include the time to harvest, quantity for packing, cost for shortage, etc. Such uncertainties can be modelled in ways other than stochastic programming. For example, Miller et al. [61] used a fuzzy mathematical program to optimize the fresh tomato packing schedule for a distributor.

Fuzzy programming was used by Zimmermann [110] to "soften" requirements by "fuzzifying" perception-based uncertainties. According to these authors, the objective function in the linear programming model could be cost minimization, while in the fuzzy model, it could be the overall satisfaction of the manager aggregated from measurements of individual satisfaction of operating costs. Although the operating costs of the fuzzy linear model were higher than in the linear programming model, they were under the budget limit. The authors summarized that their fuzzy approach had high potential to apply, to optimize, or to make general decisions, in the fuzzy environment.

A crop planning model considering uncertainties for agricultural management was presented by Itoh et al. [68] in which a linear programming model was formulated to maximize profits. However, since profit coefficients for agricultural products cannot be constant due to weather fluctuation, their linear programming model could not correctly account for environmental elements. Therefore, the authors tried to incorporate some uncertain (stochastic) parameters with fuzziness and randomness.

Mathematical models for the fresh fruit supply chain need to take perishability into account. For example, the prices of fresh fruit are at their highest during harvesting and decrease exponentially with time as products are refrigerated to reduce deterioration. Blackburn and Scudder [76] suggested a modelling framework in this category to deal with supply chain strategic problems for perishable agricultural products, such as melons and sweet corn. In their approach, the design of a supply chain for these perishable products was based on interesting questions such as: (a) how to control the timing of production including planting, harvesting and processing, (b) how to manage the ripeness time of products, and (c) how to preserve product quality through the rest of the chain. Their model separates the supply chain into two essentially independent phases: the first phase called "responsive" in which the product deterioration rate is high, and the second one is the "efficient" phase that can slow down deterioration rates. By introducing a marginal value of time (MVT) parameter, the authors showed that the appropriate method to minimize the lost value in the supply chain is a sequential combination of a responsive model for cooling fresh produce and an efficient model to reduce costs. They also pointed out there was a loose linkage between these two segments of the supply chain, and profit maximization requires close coordination throughout the chain. Their models could also be applied to the other agricultural products, such as flowers and seafood, where the time-value relationship patterns are the same as in the melon supply chain.

Another approach for perishable production is described in Amorim et al. [85] who developed multiobjective models integrating production and distribution planning with freshness considerations, such as fixed shelf-life and loose shelf-life. In each case of freshness, both integrated and decoupled models were proposed to be compared with achieving the bi-objectives of (1) minimizing total costs and (2) maximizing delivered shelf-life products. Depending on the type of shelf-life of the perishable products or the functional area decision making (production or distribution), the authors developed a multiobjective mixed-integer linear model for the fixed shelf-life, and a multiobjective mixed-integer non-linear model for loose shelf-live because of uncertain nature of it. To compare the differences between the integrated and decoupled models for fixed and loose shelf-life, a case study was used. The results showed that the integrated models solved the problem better than the decoupled ones. In addition, the loose shelf-life models could be implemented to deal with the randomness of the spoilage process.

Hester and Cacho [67] had suggested a hybrid model with an objective of maximizing the net present value for apple orchard systems by optimizing pruning plans over a 15-year period. Their approach presented a combination of multiple approaches: dynamic optimization, genetic algorithm (GA) and nonlinear programming (NLP). A dynamic simulation was developed, based on the interactions of the complex biological and economic relationships in apple orchard networks. It can be used for the managers of the apple orchards to consider problems including how apple tree yields could be influenced by biological factors, and how to choose apple variety among a large number of apple orchards. In each apple orchard, various tasks such as selecting varieties, grafting roots, identifying intervals between trees, training planting methods can influence yield, quality of apples harvested, as well as earned profits.

Another type of model was conceived by Bezat-Jarzębowska and Rembisz [35] to support agri-food producers to maximize expected profits by optimizing production where the scale of production does not impact competitive markets. The stochastic frontier approach (SFA), which is commonly used economic modelling primarily dealing with high randomness, had been used in their model. The SFA was the combination of two functional forms including the input(s)−output relations (the Cobb−Douglas' model [111]) and a translogarithmic model to improve the efficiency of the farm production, or in other words, the profitability was increased. The model was implemented and validated by a data set collected by Farm Accountancy Data Network in Poland.

A machine repair model was modified and applied by Ampatzidis et al. [36] to optimize fruit harvesting and bin loading. Inefficient harvesting and postharvest activities, increased costs were identified and reduced, and the schedules of labor and machinery were also improved. The authors implemented the machine repair model for two harvesting fruit processes: picking and bin loading. To adapt properly, the picking workers were considered as machine breakdowns and the fruit collection points were system servers for the picking process. Similarly, transport workers were considered the machine breakdowns and trucks (unloading points) for the bin loading process. The model was built and solved by using MATLAB to evaluate how the two-process system performed. To validate the model, two specific types of fruit were chosen as case studies: table grapes in Greece and sweet cherries in Washington State, USA. The reason why the authors chose two different fruits grown in different places is based on the difference in the size of bins and trucks.

Summary of section: Various types of mathematical models were used in the articles reviewed in this section. The authors used methods such as fuzzy logic, heuristic algorithms or nonlinear programming models. Future works should aim at combining one or more of these methods with meta-heuristics to deal with large-scale FFSC problems.

5.5 Robustness and Limitations of Existing Models

In the general view, the following are the main criteria on which researchers conceive and structure their models for the planning and logistics of the fresh fruit supply chain:

a. Relationship between price and demand
b. Environmental constraints
c. Planting/harvesting times and shipping/transporting factors
d. Operational decision-making styles

The fruit species under consideration were varied but very commonly consumed on a daily basis such as tomatoes, apples, grapes, bananas, etc. However, many tropical fruits were not covered extensively in the literature.

Figure 5.3 shows the coverage of the fresh fruit supply chain research in the past, focusing on several common species in the market.

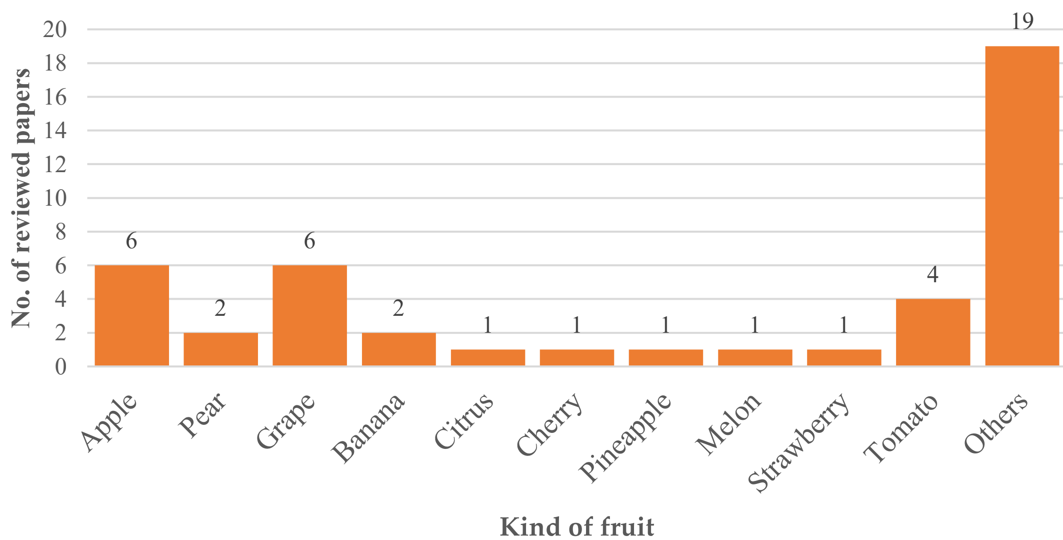

Figure 5.3 Statistics related to the fresh fruit supply chain research in terms of fruit species from 1978–2017.

It can be observed that apples, grapes and tomatoes are the most used in case studies. From the chart in Figure 5.3, it is noticeable that most cases covered in the literature are perennial fruit or one-year lifetime trees. Meanwhile, fast-growing perennial trees are less considered than one-year trees.

Figure 5.4 shows that most of the papers deal with tactical decisions (17 articles) followed by the operational level (11 articles). Only six articles focused on the strategic decision level. However, the coupled models seemed to be the most favorite approach in dealing with the fresh fruit supply chain, including 24 articles at different levels for combined decisions, such as strategic–tactical (7 articles) and tactical–operational (17 articles).

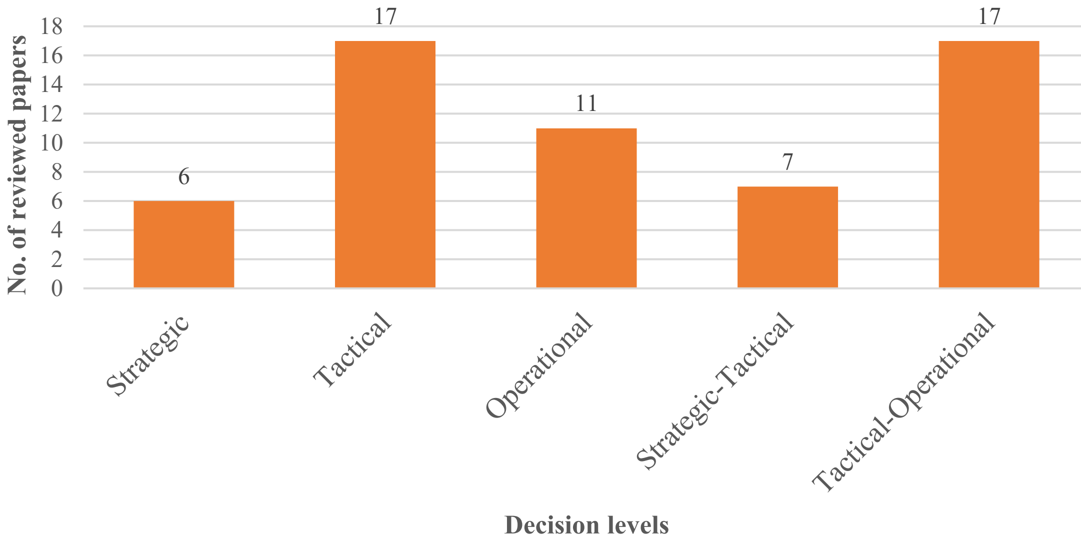

Figure 5.4 Number of model categories in fresh fruit supply chain optimization.

It was observed that all models were based on the well-defined underlying factors such as links, actors, actions, and states. The product is characterized by two types of states: appearance and quality. Appearance states are affected by handling action while the quality states are affected by processing, transportation and storage.

The models in the literature can be viewed from a robust point of view [112]. The linear programming models include crop growth characteristics, seasonality, weather data, crop growth in order to accurately model yield and the timing of the maturity of fruit. The model such as Hamer's [21] can also be used several times during a season to update marketing strategy and early warnings of production surpluses or shortfalls. However, linear programming models are limited because they do not consider uncertain parameters. In theory, linear models can be run for a choice of parameters but organizing such sensitivity analyses in a risk/robustness/economic trade-off is a challenge.

The dynamic programming models are useful in incorporating the quality. The process of changing the quality of products was described as one function of the DP model developed by Gigler et al. [66][30]. In addition, the appearance of the products was also considered in their DP approach to point out the relationship between quality and appearance of the products through the supply chain from farmers to final consumers.

The "machine repair" model suggested by Ampatzidis et al. [36] could be part of a general simulation approach to integrating operation research techniques to improve the total harvest process. It can also be used to model machine harvesting with a fleet of machines to collect, pack and distribute fruits. Consequently, the queuing theory to model waiting time can be logically used for agricultural operations.

Summary of section: This section presented the distribution of papers according to the type of fruit considered and the decision planning levels. Future models should aim to incorporate biological, environmental, and economic considerations such as epidemics, climate change, and price fluctuations. Furthermore, more emphasis should be put on fast-growing perennial trees and their fruit production and distribution.

5.6 Conclusions and Research Perspectives for Future Models

The fresh fruit supply chain, a subcategory of the fresh product supply chain, presents modelling challenges due to inherent random and uncertain factors (such as yield, demand, price). New information technologies associated with robust but affordable computers and high technologies (such as drones and sensors) could allow FFSC managers to monitor real-time crop growth information to develop better harvesting and production plans. Nevertheless, according to our review, there exist three main challenges for FFSC models: (1) the efficiency of the entire FFSC in function of the coordination between different stages; (2) the development of integrated planning models which are capable of acquisitioning data or updating parameters from such high-tech informative systems; and (3) surprisingly, a lack of standardization of all FFSC model outputs and performance metrics is observed. FFSC modelling is an interdisciplinary topic and communication between disciplines should be improved to facilitate model comparisons. These shortcomings can hinder the adoption of modelling tools by practitioners. Besides, social responsibility and changing consumer values will create an increasingly complex business planning environment. Examples of such issues are genetically modified products and organic fresh fruits. The creation of a database of FFSC models, categorized by local and regional factors, would be valuable for modelling practitioners and modelers, allowing benchmarking to occur.

Sustainable development and sustainability criteria have become extremely controversial since international trade based on economic criteria is seen as increasing green-house gas emissions and creating waste management issues. At the same time, increasing global competition and lower prices are strongly required for efficient management, including well-organized transportation, distribution, and inventory management of fresh fruit. These are essential for profitability and provide additional research opportunities.

All models reviewed are appropriate for the specific context and problems that they deal with. Of all the papers reviewed, the following propose integrated models of the supply chain under:

- deterministic conditions (Hamer, 1994 [21], Munhoz and Morabito [30])
- stochastic contexts (Bezat-Jarzębowska and Rembisz [35], Ahumada et al. [33])
- sustainability considerations (Foong et al. [99], Van Der Vorst et al. [77])
- multiobjective optimization (Cittadini et al. [75], Soto-Silva et al. [97])
- multistage and multiechelon networks (Darby-Dowman et al. [31], Trivedi et al. [107])
- comprehensive case studies (Broekmeulen [63], Verdouw et al. [81])

For future models, there are two facts that we cannot ignore: (1) mechanization in all steps of the fresh product supply chain and (2) door-to-door service which is becoming prevalent worldwide, especially in periods of pandemic. Mechanization is a response to the combination of rising labor costs and increased opportunities for rural workers in nonfarming sectors. Information and communication technologies in association with automation systems (data-driven technologies, artificial intelligence, etc.) can replace manual decision making in the traditional farm. FFSC systems then become more complex, requiring higher investment. This creates new opportunities for mathematical modelling. Therefore, there is a strong need for models that include real-time monitoring data, uncertain information, logistics integration and product safety and quality. The extension of current models to incorporate robustness and risk reduction would be extremely useful.

Regarding the door-to-door service, it is becoming prevalent in many Asian countries and the North American market. To meet a client's specific requirements, the retail and food service supply chains must evolve various services, including customs formalities, preparation of space for perishable cargo on transportation means, prioritization of storage at places of origin and destination. Factors such as temperature, quantity, damaging degrees, sanitary inspections and quality checks are standard and need to be regularly controlled. The big question is how we could insert all these features into current optimization models.

CHAPTER 6 DETERMINISTIC OPTIMIZATION MODEL FOR DRAGON FRUIT PLANTATION PLANNING

This chapter is based on the article "Optimization Model for Fresh Fruit Supply Chains: Case-Study of Dragon Fruit in Vietnam" which appeared in *AgriEngineering*, 2(1), p. 1-26 [41]. For more details, please refer to the electric copy that has been presented at https://www.mdpi.com/2624-7402/2/1/1.

6.1 ABSTRACT

We present an optimization model for dragon fruit plantations in Vietnam. The timing of cultivating and harvesting decisions are taken into account as the dragon fruit plant has an approximately ten-year life cycle with maximum average yield in the fourth year. Another consideration also included is the prevalence of forward-buying contracts with locked-in prices. The dragon fruit supply chain faces several difficulties as yield, price, and demand are highly sensitive to weather conditions and global uncertainty factors. The risk factors in the dragon fruit supply chain also depend on species—for example, the red varieties, while more profitable than the white varieties, also have higher export risk because they are subject to global prices and adverse geopolitical conditions.

6.2 INTRODUCTION

In recent years, along with the development of the supply chain management and logistics industry, the agri-food supply chain in general and the fresh fruit chain in particular have been recognized as strategic components of the national economy of many developing countries such as Vietnam.

Compared to staple crops, fruit production brings greater economic benefits. However, the fruit and vegetable production sector also faces particular risks such as climate change, water scarcity, increase in land-use for industrial and urban development, and consumer behavior and price volatility. Moreover, the planning and managing of production, distribution, and pricing of fresh fruits are more complicated because of their very short shelf-life.

The value can be increased if the value chain of fruit and vegetable production and distribution is better organized from farmers to retailers. Countries where agriculture is in development, i.e., Vietnam, are still facing challenges such as:

- The influence of traditional trade practices—there are many intermediate nodes involved in the network, making the food supply chain longer and more complex than in other developed countries.
- The high cost of storage after harvesting and transportation - this is due to the tropical climate with high temperature and humidity.
- The continued use of low paid labor. Though labor is cheap, there is a high workforce turnaround. Workforce shortages are acute at the beginning and end of the harvest season when labor demand is high due to competition. During these periods, workers often change employers for better pay.
- The poor availability of information within the value chain from growers to collectors/traders, wholesalers, retailers, and supermarkets about the harvest, preliminary processing, packing, labeling, preserving, and transportation.
- The inability of farmers to set produce prices—farmers play the most important role in the food supply chain but most of them are small, with little influence on price. They must sell their products at prices determined by traders due to lack of market information and experience.
- Due to the lack of long-term orientation at the macro-level of management, farmers target profits based on market demand. In the case of dragon fruit, this may imply cutting existing varieties of the fruit and changing over to other varieties, based on anticipated demand. Since dragon fruit is a perennial plant, the impact of these decisions can last several years.

It can be said that making decisions for a fresh fruit supply chain management is a more difficult and complex problem than with other supply chains [50]. This has been a great challenge for fresh fruit supply chain (FFSC) managers over the past 40 years, given the increasing globalization and rapidly increasing demand. They need a tool to support modern and accurate decision-making for long-term production. There are several articles in the literature related to FFSCs with many different approaches or methods that could support optimization of a part or the whole chain. The deterministic approach is a very common and often used in the FFSC research; formulations are based on both linear programming [20-24, 68] and mixed integer programming [25-28, 71, 85, 89, 95, 96, 113].

There are two essential types of fruit used for case studies: perennial crops such as apples [34, 55, 56, 67], oranges [20, 96], pome fruit [89], pears [25, 72, 93], cherries [75] and grapes [24, 69, 78] or annual crops such as pineapple [77], strawberries [86], melon [76], and tomato [26, 27, 33, 61]. Dragon fruit is a tropical fast-growing perennial crop, other examples being asparagus [114], Persian lime [90], Thai soursop, Taiwan pear-shaped guava, etc.

During the last several years, the fresh produce cold chain has received attention from researchers around the world to enhance the quality and freshness of fruits and vegetables delivered to customers. The cold chain issues considered by most researchers have to do with controlling the temperature and gas flow in containers [115-117], minimizing the energy used to refrigerate containers [118], and optimizing the transport system in the chain [119].

The dragon fruit (Figure 6.1) is a tropical fruit grown extensively in Vietnam. With 36.5 thousand hectares of cultivated land and 630 thousand tons of total yield, Vietnam is the world's leading exporter of dragon fruit [120]. However, dragon fruit production and processing are still in a nascent stage of development and face issues around severe price fluctuations due to conditions such as:

a. Product development is still nascent.
b. Market price fluctuations.
c. Chinese imports are subject to price and currency exchange risks.
d. High competition with other exporting countries (such as Thailand, Malaysia, etc.) driving down value despite increased export volumes.
e. Exports have been increasing both in volume and value but the increase in value has been declining.

This chapter presents an optimization model for dragon fruit crop planning to support farmers making decisions on the allocation of land to crop varieties. The objective of the model is to maximize profit while satisfying customer demand. Given that dragon fruit is perennial but fast growing, there is an opportunity to change the crop mix based on anticipated future prices. However, there could be a loss in yield depending on the maturity of the crops in a plantation mix. The remainder of this chapter is organized as follows: Section 2 introduces dragon fruit plantation and crop planning. Section 3 presents a linear programming optimization model for crop harvesting and replantation decisions. Section 4 presents the results and discussions from example scenarios. Section 5 concludes the chapter and outlines areas for further study.

Figure 6.1 Dragon fruit trees blossoming (left) and fruiting (right) in June–July. Photos taken in 2016 in Binh Thuan province, Vietnam

6.2.1 Fruit Distribution Context

The dragon fruit supply chain starts with farmers who make plantation decisions based on forward buy-in contracts with traders. The traders sell fruit to by-products and wholesalers who in turn distribute the fruit and by-products to retail, export, and by-product producers. The dragon fruit supply chain [121] is depicted in Figure 6.2. Typical dragon fruit by-products are wine and packaged dried fruit snacks.

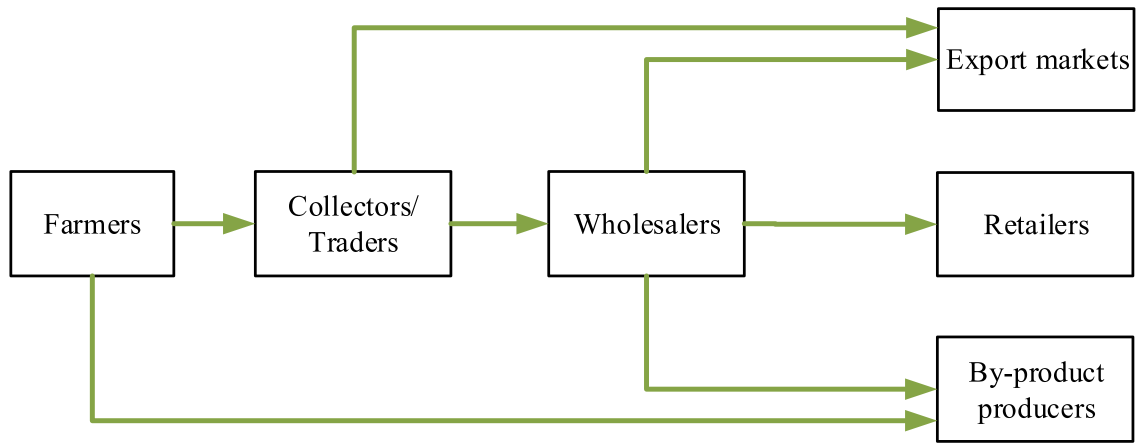

Figure 6.2 Simplified dragon fruit supply chain.

Dragon fruit is typically harvested twice a year in Vietnam. Harvesting starts one year after plantation, but the fruit is at the quality required for commercial purposes 2–10 years after harvest. Dragon fruit yield typically depends on age and a tree is usually only considered productive until the age of 12 years. Figure 6.3 shows the typical yield curve as a function of tree age.

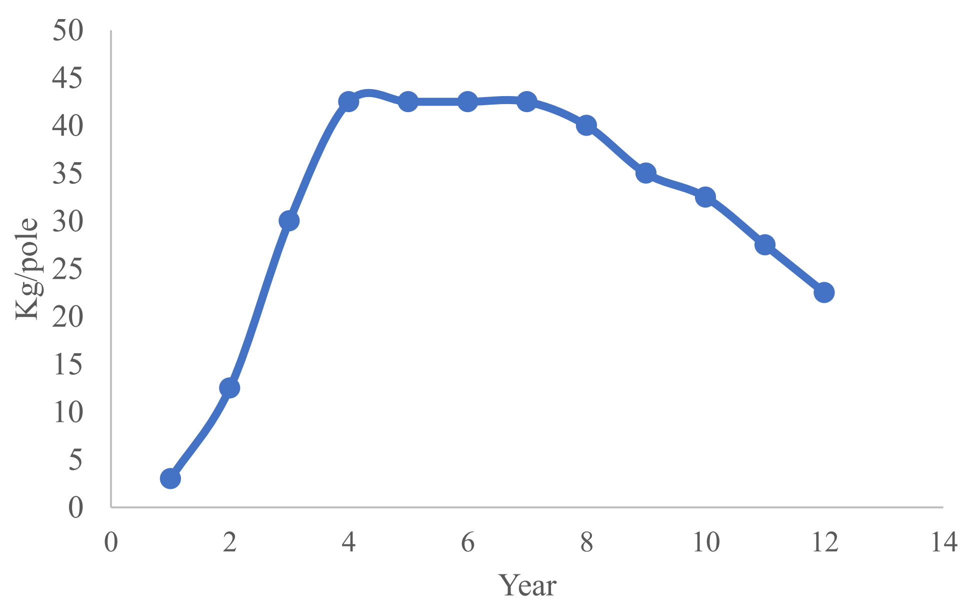

Figure 6.3 Typical dragon fruit yield as a function of tree age.

There are three varieties of dragon fruit planted in Vietnam: the red-skin white-flesh, the red-skin red-flesh and the yellow-skin white-flesh (Figure 6.4). The red-skin varieties are very popular, the white-flesh variety being the most sold. The red-skin red-flesh variety has a high demand during the Lunar New Year and is also exported extensively to China. The yellow peel white flesh variety is relatively new in Vietnam and is available only in major metropolitan areas.

Red-skin white-flesh Red-skin red-flesh Yellow-skin white-flesh

Figure 6.4 Species of dragon fruit planted in Vietnam.

The dragon fruit blooms from May to August and is ready a month later for harvesting in September and October. However, dragon fruit prices are usually low in the main season (due to the availability of the fruit). Due to its high economic value in January and February, dragon fruit growers install lighting systems to stimulate trees to bloom and have fruits to improve productivity in the dry season which lasts from November to April [121]. Therefore, there are two times for harvesting, either from May to October (rainy season or season 1) or from November to April (dry season or season 2) (Figure 6.5).

May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr
Rainy season (season 1)						Dry season (season 2)					

Figure 6.5 Dragon fruit production calendar in Vietnam.

Dragon fruit trees are commercially viable for 10–12 years, but because they grow quickly, the plantations can continue to harvest existing crops below that age or cut them down for investment in other varieties based on demand and price.

6.3 METHODOLOGY

The methodology in this chapter is aligned with the hierarchical planning approach which separates the decision-making process into tactical and operational phases [37]. In hierarchical planning, decisions are first made at the tactical level and then at the operational level. Figure 6.6, which is adapted from Ahumada et al. [33], shows how the hierarchical approach may be applied to the dragon fruit chain.

In this chapter, a quantitative modelling approach for decision making for dragon fruit plantation and harvesting in Vietnam is presented. As previously mentioned, this approach looks at planting and cutting (which are tactical decisions) taken over a multi-year planning horizon. The potential benefit from the hierarchical planning is that growers can be involved in making decisions about the market and production. In other words, coordinating tactical and operational decisions is beneficial for multiple parties: growers, producers, distributors, and vendors.

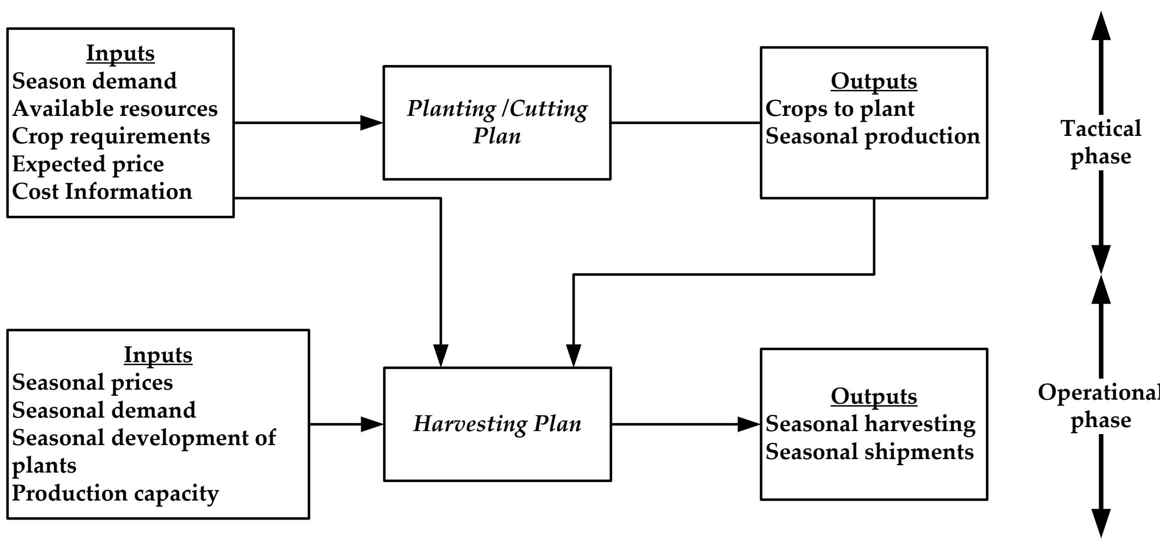

Figure 6.6 Example of a hierarchical planning schematic.

In Figure 6.6, the first phase deals with tactical decisions that are only made at the start of the season such as what crop to plant or truncate, and when and how much to plant or truncate. While in the second phase, farmers must decide how much to sell to customers in each season according to the market conditions.

To execute the hierarchical planning schematic, a deterministic optimization model is developed in this chapter for the dragon fruit production in Vietnam. Although dragon fruit is used as the main target of the current study, our model can certainly be adapted to other fresh fruit production chains.

6.4 LINEAR PROGRAMMING OPTIMIZATION MODEL

6.4.1 Hypotheses and Assumptions

For this model, it is assumed that:

a. The facilities (farms) are already operational since the model does not deal with network decisions.
b. The time horizon for tactical planning is 10 years and two harvesting seasons (rainy season and dry season) for each year are considered.
c. Fruit trees are cut down when they are 10 years of age or earlier, if the model chooses to (e.g., when the future prices of other crops are much higher than the planted crop). It is assumed that cutting down or replanting decisions are only carried out in season 1.
d. The distribution of yield, demand, and market prices are represented by their expected values.
e. Storage is not allowed for fresh dragon fruit. However, the fruit may be used for by-products such as dry snacks and wine.
f. The amounts of each crop to cut and plant are decision variables which cannot exceed the maximum amount of land that is already determined.
g. Decision variables of the model are the new plantation and truncating areas of each crop, and the amount of fruit sent to customers (traders, wholesalers, and by-product producers) every year.
h. Other decisions include the quantity of fruit to sell to customers and the amount of labor (fixed and part-time) required to cover all the activities in the model.

6.4.2 Objective Function

$$\begin{aligned}\text{Max } O = &\sum_j\sum_i\sum_s\sum_t p_{jist}.ST_{jist} + \sum_j\sum_m\sum_s\sum_t q_{jmst}.SWM_{jmst} \\ &+ \sum_j\sum_s\sum_t r_{jst}.SB_{jst} - \sum_j\sum_t cp_{jt}.Y_{jt} - \sum_j ch_t.X_{jkt-1} \\ &- \sum_j cr_t.Z^1{}_{j,k=10,t} - \sum_t F_t.cLabf - \sum_t Hire_t.cLabp \\ &- \sum_j cbp_{st}.SB_{jst} - \sum_t\sum_s\sum_j cPNT_{jist}.\epsilon_{ijst} \\ &- \sum_k\sum_s cwater_{ks}.w_{ks} - \sum_j\sum_{s=2} clighting_{js}.v_{js}\end{aligned} \tag{1}$$

The objective of the proposed model is to determine the planting and harvesting decisions that maximize expected profit for the farmers; this is the difference between the total revenue expected from selling to traders (ST), wholesale markets (SWM), and by-product suppliers (SB), and the total costs of planting, truncating, by-product processing, penalty for missing demand, labor, lighting cost, and watering. The notation is presented in Appendix B1.

6.4.3 Constraints

The first category of constraints (2, 3, and 4) is related to resources (land, water, and lighting).

- Land availability

$$\sum_j\sum_k X_{jkt} \leq \text{L} \qquad \forall t \tag{2}$$

Total area of each crop j at age k cannot exceed the available land (L)

- Water restriction

$$\sum_j\sum_k X_{jkst}.w_{jkst} \leq W_{st} \qquad \forall t, s \tag{3}$$

At each tree age k, the amount of water w required per hectare for each crop j is different in season s of year t; this cannot exceed the availability of water. Also, yield decreases if the trees are watered too much.

- Lighting restrictions

$$\sum_{j} X_{js}.v_s \leq V_s \qquad \forall s \tag{4}$$

Due to lack of sunlight in the dry seasons ($s = 2$), a light supplementing method is applied at night. At each tree age k, the requirement of light per hectare for each crop j is different but limited.

Constraints (5) and (6) are for plantation area and yield.

- Minimum plantation size

$$\sum_{j} X_{jtk} \geq u_{jt} \qquad \forall t \tag{5}$$

Constraint (5) is the lower bound for the planting area for each crop in a given period, which may come from forward contracts. The minimum planting area of each crop u is defined by the planner depending on commitments to customers. The parameter u could be 0 but should be less than the available land L.

- Yield

$$\sum_{i} ST_{jist} + \sum_{m} SWM_{jmst} + SB_{jst} \leq \sum_{k} \gamma_{jkst} X_{jkst} \qquad \forall j, s, t \tag{6}$$

Constraint (6) ensures that the total harvest is less than the yield (metric tons per hectare) times plantation area (in hectares).

- Plantation age class balance

The third category of constraints (7–15) is related to the planning structure for agriculture models. The cutting down and replanting of new varieties has been modeled in Catalá [89] for a case study on apple and pear trees.

$$X_{j,k,s=1,t} = X_{j,k,s=2,t} \qquad \forall j, k, t \tag{7}$$

Constraint (7) ensures that there is no change in the planted area of each crop within a year (i.e., between season 1 and season 2). This is because plantation or truncation decisions are only made at the start of season 1.

The plantation decisions are decided by Y_{jt}. The age of a newly planted fruit tree is always 0. There are two types of truncations: ${Z^1}_{jkt}$, which is optional for a tree of age $k = 1..9$ and ${Z^2}_{j,k=10,t}$ which is mandatory for all trees that have reached an age of 10.

$$X_{jkst} = Y_{jt} \qquad \forall j, k = 1, t = 1 \tag{8}$$

Constraint (8) states that in year 1 only new crops (age class 1) can be planted.

$$X_{jkst} = I_{j,k-1} - {Z^1}_{jkt} \qquad \forall j, k = [2..9], t = 1 \tag{9}$$

Constraint (9) is similar to constraint 8 and applies only to year 1 but for other age classes (*k*=2..9). It states that the plantation area is the inventory of trees of age class *k*-1 in year 0 less what can be cut down in year 1 after they have aged by 1 year.

$$X_{jkst} = I_{j,k-1} - {Z^2}_{jkt} \qquad \forall j, k = 10, t = 1 \tag{10}$$

Constraint (10) states that age-10-crops that have to be cut down in year 1 while determining the initial plantation area.

$$\sum X_{jkst} = \sum Y_{j,t-1} - \sum {Z^1}_{jkt} \qquad \forall j, k = 1, t > 1 \tag{11}$$

Constraint (11) states that for periods $t > 1$ in the planning horizon, the plantation area for age class $k = 1$ is determined by new crop planted the year before less whatever is cut from that new plantation the next year.

$$X_{jkst} = X_{j,k-1,s,t-1} - {Z^1}_{jkt} \qquad \forall j, 10 > k > 1, t > 1 \tag{12}$$

Constraint (12) is for crop ageing for age classes $10 > k > 1$. The plantation size in a given year depends on what it was the previous year, less the area cut down optionally.

$${Z^2}_{jkt} = X_{j,k-1,s,t-1} \qquad \forall j, k = 10, t > 1 \tag{13}$$

Constraint (13) states that all crops of age 9 in a given year *t*-1 should be cut next year.

- Labor constraints

$$F_t + Hire_t - P_t.\sum_j Y_{jt} - H_t.\sum_j X_{jt} - R_t.\sum_j Z_{jkt} = 0 \qquad \forall t \tag{14}$$

Constraints (14) models workforce requirements to plant, cut, and harvest in given year.

$$F_t = M \qquad \forall t \tag{15}$$

The number of full-time workers is sometimes a fixed number. If that is the case, the full-time complement of workers should be set to that number.

$$Hire_t \leq N \qquad \forall t \tag{16}$$

The number of part time workers hired based on requirements of cultivating or harvesting or truncating. However, the number is limited due to budget, as seen in constraint (16).

The last set of constraints, (17) to (19), is for demand satisfaction:

$$\sum_j ST_{jist} = d_{jist} - \epsilon_{jist} \qquad \forall t, s \tag{17}$$

Constraint (17) is a soft constraint on trader demands, given that under-shipping to them is allowed.

$$\sum_j SWM_{jmst} = e_{jmst} \qquad \forall t, s \tag{18}$$

Constraint (18) states that the demand of wholesalers should be satisfied.

$$\sum_j SB_{jt} = f_t \forall t \tag{19}$$

Constraint (19) states that the demand for by-products should be satisfied.

6.4.4 Definition of indices, variables, and parameters of the deterministic model

Indices:

t	Time periods
k	Age classes in the plantation, each representing a two-year period
s	Harvesting season (1 for wet, 2 for dry)
j	Different species of dragon fruit
i	Traders
m	Wholesale markets (WM)
b	By-product

Parameters:

L	Amount of land available
w_{ks}	Water required per hectare for crop *j* of age class k in season s
v_{js}	Lighting required per hectare for crop *j* in season *s*
W_s	Water restriction in season

V_s	Lighting restriction in season
u_{jt}	Minimum planting area per crop *j* in period *t*
γ_{jkst}	Yield in kgs per hectare of crop *j* belonging to age class k in season s
p_{jist}	Price per kg of crop *j* for trader *i* in season s of period *t*
q_{jmst}	Price per kg of crop *j* for wholesaler m in season s of period *t*
r_{st}	Price per kg of byproducts (e.g., wine) in period *t*
d_{ijst}	Demand of trader *i* for crop j in season s of period *t*
e_{mjst}	Demand of wholesale market *m* for crop *j* in season s of period *t*
f_t	Demand for byproducts (e.g., wine) in period *t*
P_t	Number of workers needed to plant one hectare
H_t	Number of workers needed to harvest one hectare
R_t	Number of workers needed to cut one hectare
M	Maximum number of fixed workers in a period
N	Maximum number of part-time workers in a period
I_{jk}	Initial area of crop *j* of age class *k*

Cost parameters:

cp_t	Cost per hectare of planting in period *t*
ch_t	Cost per hectare of harvesting in period *t*
cr_t	Cost per hectare of cut in period *t*
cbp_t	Cost per kg of processing (e.g. wine)
$cLabf$	Cost of fixed workers per period
$cLabp$	Labor cost of part-time workers per period
$cPNT_{jist}$	Penalty for not meeting demand per kg of crop *j* for trader *i* in season s of period *t*
$cwater_{ks}$	Cost of required water per hectare for crop *j* of age class *k* in season *s*
$clighting_{js}$	Cost of required light per hectare for crop *j* in season *s*

Variables:

X_{jkt}	Plantation area of crop *j* in period *t* of age class *k*
ST_{jist}	Quantity of crop *j* shipped to trader *i* in season s of period *t*
ϵ_{ijst}	Quantity of crop *j* under shipped to trader *i* in season s of period *t*
SWM_{jmst}	Quantity of crop *j* shipped to WM *m* in season s of period *t*
SB_{jst}	Quantity of crop *j* harvested for by-products (e.g. wine) in season s of period t
F_t	Number of fixed workers in *t*
$Hire_t$	Part-time workers hired in period *t*
Y_{jt}	Area of crop *j* planted in period *t*
${Z^1}_{jkt}$	Area of crop *j* of age class *k* cut optionally in period *t*
${Z^2}_{jkt}$	Area of crop *j* of age class *k* = 10 that must be cut in period *t*
Z_{jt}	Area of crop *j* of age class *k* cut in period *t* in total

6.5 CASE STUDY

The model for dragon fruit cultivation presented in the previous section was applied using the conditions of an actual dragon fruit plantation in Vietnam. The authors of this study contacted a small growing operation covering an area of about 20 hectares. Data were obtained on dragon fruit prices and demands, planting, replanting, and harvesting costs, labor availability, water and light requirements, species yields, etc. The important issue facing such operations is land management, where farmers need to make decisions on land allocation for different species of dragon fruit over a period of 10 years. The model is intended to allow farming communities to evaluate alternative land allocation and commitment scenarios based on different prices.

As mentioned, farmers are planting two kinds of dragon fruit (white-flesh and red-flesh) on their lands. The price of each crop is different and depends on market demands. The price of the red-skin white-flesh dragon fruit (Crop 1) is stable in season 1 (favorite season) and increases a bit in season 2 (off-season). The price of the red-skin red-flesh dragon fruit (Crop 2) is double because Crop 2 is planted for export to China, where the demand is always good. However, the price of this crop fluctuates highly, relying heavily on Chinese traders. Verbal agreements are usually made between farmers and traders; however farmers could be ruled by the prices set by the traders [121]. In the case of traders cancelling deals, the price drops dramatically.

The yellow-skin white-flesh dragon fruit (Crop 3) (currently imported from Malaysia) has only recently appeared in the market and has had an extremely high price for the last three years. It is still in great demand because of its sweetness and the curiosity of consumers. Farmers intend to grow Crop 3 trees to cover that demand but have some disadvantages: the plants are novel and disease prone, and yields are just one-third of Crop 1 or Crop 2 (according to the experience of many farmers). However, farmers like to grow all three kinds of dragon fruit, to cover all market demands and hedge their risks against demand and price.

In the baseline scenario, the Crop 1 is the traditional chain for both domestic and export markets with a stable demand, while Crop 2 is only planted for export to China. Crop 3 is cultivated only for the purpose of testing its viability in the consumer market. The model assumes that the demands and selling prices of all crops increase steadily over 10 years.

The land proportion for each type of dragon fruit is proposed to help farmers managing their costs and benefits in planning the combination framework to install various dragon fruits, and also to decide thereafter which dragon fruit category they have to grow within 10 years. The mean values for one year are based on the previous year's data. Using recent pricing in the Vietnamese market it was found that: a) the price of Crop 3 is around 10 times higher than the price of Crop 1, and b) the price of Crop 2 is three times as high as for Crop 1.

After the baseline scenario is completed and analyzed by the proposed model, other scenarios are developed, based on the minimum limit of area for each crop of dragon fruit to plant, and the fluctuations of the prices and market demands. To test how the model adapts to any changes or requirements of dragon fruit production, groups of different expansion scenarios are developed with various assumptions. The model is implemented using open-source LP/MIP solver GLPK/GUSEK, that was developed by Free Software Foundation, Inc., Boston, USA, and it is computationally tractable (Appendix C1).

6.5.1 Baseline Scenario

In this scenario, approximately 15 hectares of Crop 1 and Crop 2 have been planted on 20 hectares of land to meet orders in year 0 (the start of the planning horizon). All initial input values such as yields, prices, demands, labor costs, and resource costs were collected from farmers, market reports, and dragon fruit cultivating guidelines in 2017. Figure 6.7 shows the typical yield curve of three dragon fruit varieties as a function of tree age [12, 122].

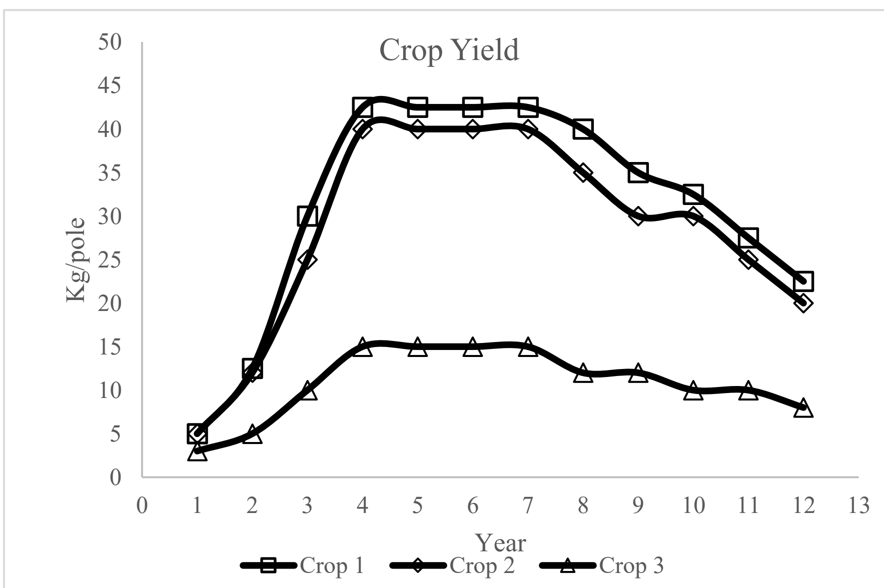

Figure 6.7 Yields of three dragon fruit varieties.

It is assumed that all dragon fruits are sold to five traders and five wholesalers. After meeting the needs of all traders and wholesalers, if any amount of Crop 1 is left over, it is sold as a by-product to produce wine and snacks. General information of average yields, demand, and prices of each crop in the current market [122] is show in Table 6.1:

Table 6.1 General information for the model

	Average Yield	Average Demand	Average Price
Crop 1	15	30	0.5 US$
Crop 2	14	30	1.5 US$
Crop 3	5	5	5 US$

The results of the baseline scenario are shown below:

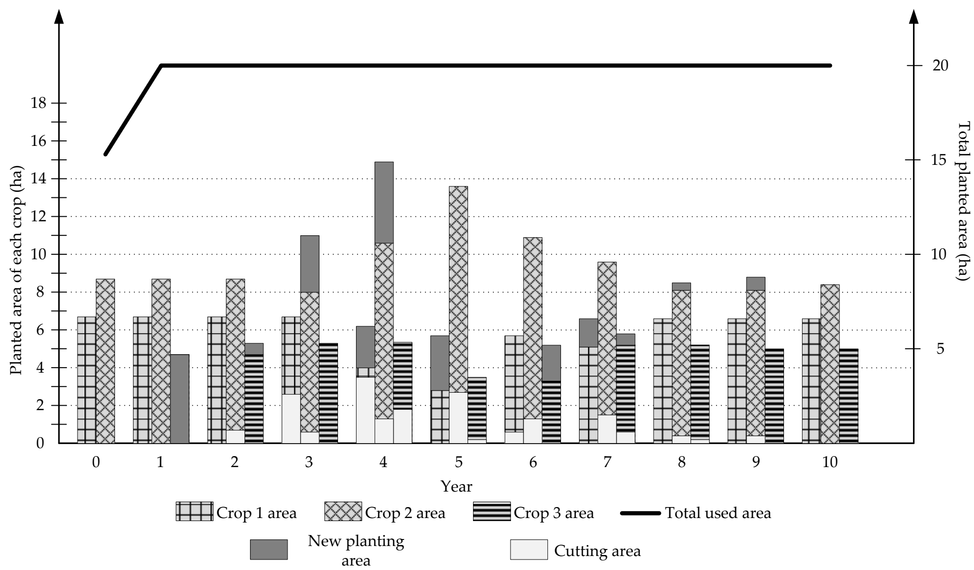

Figure 6.8 Baseline model result.

Figure 6.8 shows the recommended dragon fruit cultivation allocation to crops on 20 hectares of land over 10 years. Crop 3 is planted primarily to meet demand; the land area allocation of Crop 1 and Crop 2 are relatively constant through the 10 years. We can see that growing new trees and cutting old ones occur on a large area of land from year 3 to year 6; this also affects the profitability of the farmers. The profit increases rapidly in the first three years due to income from Crop 3. However, it goes down in year 4 because many older Crop 1 and 2 plants are truncated and replaced by new ones. The variations in revenues, profits, and costs are shown in Figure 6.9. Figure 6.10 shows the profit of each crop over 10 years.

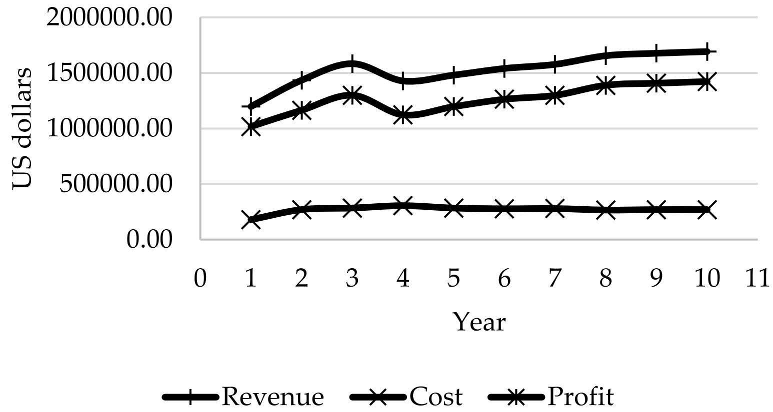

Figure 6.9 Revenue, profit, and cost of the baseline scenario for the 10-year horizon.

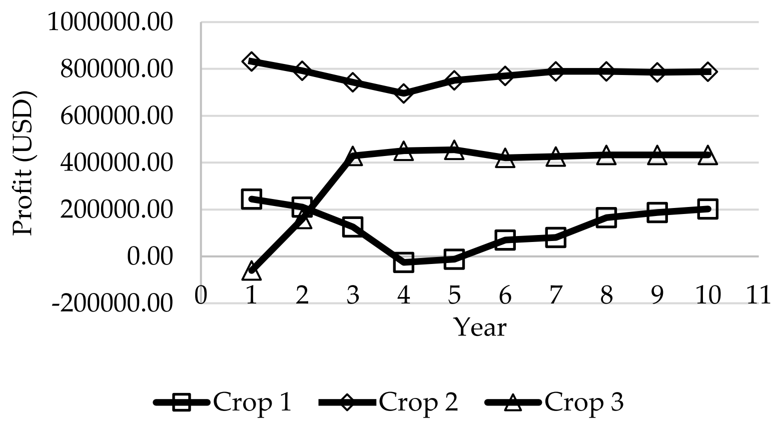

Figure 6.10 Profit of each crop.

Building upon the baseline scenario, four different expansion scenarios are proposed with assumptions about changes to prices or demands, price fluctuation with a probability factor, and the initial plantation. All scenarios are described briefly in Appendix B2 (Table B1).

6.5.2 Changes to Price of Crop 2

In this scenario, the price of Crop 2 is considered increasing or decreasing linearly within the range of 0.5 US$ (Crop 1 price) and 5 US$ (Crop 3 price). The other information is the same as in the baseline scenario. The cultivation changes for each crop when the price of Crop 2 changes in two cases are shown below:

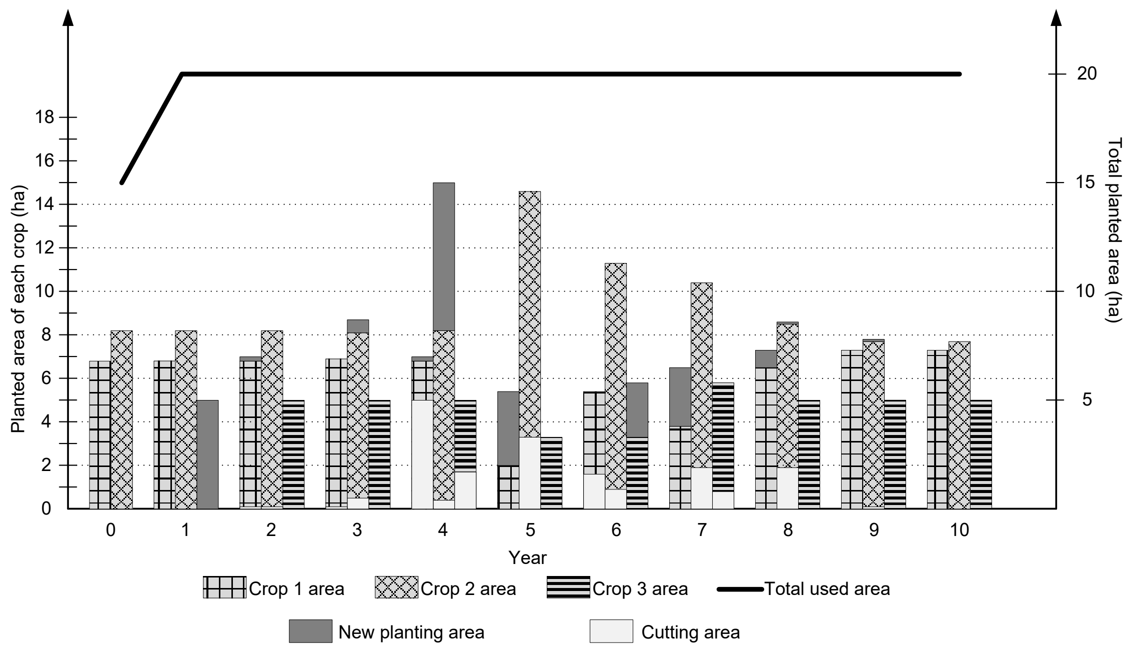

Figure 6.11 Plantation allocation when Crop 2 price increases.

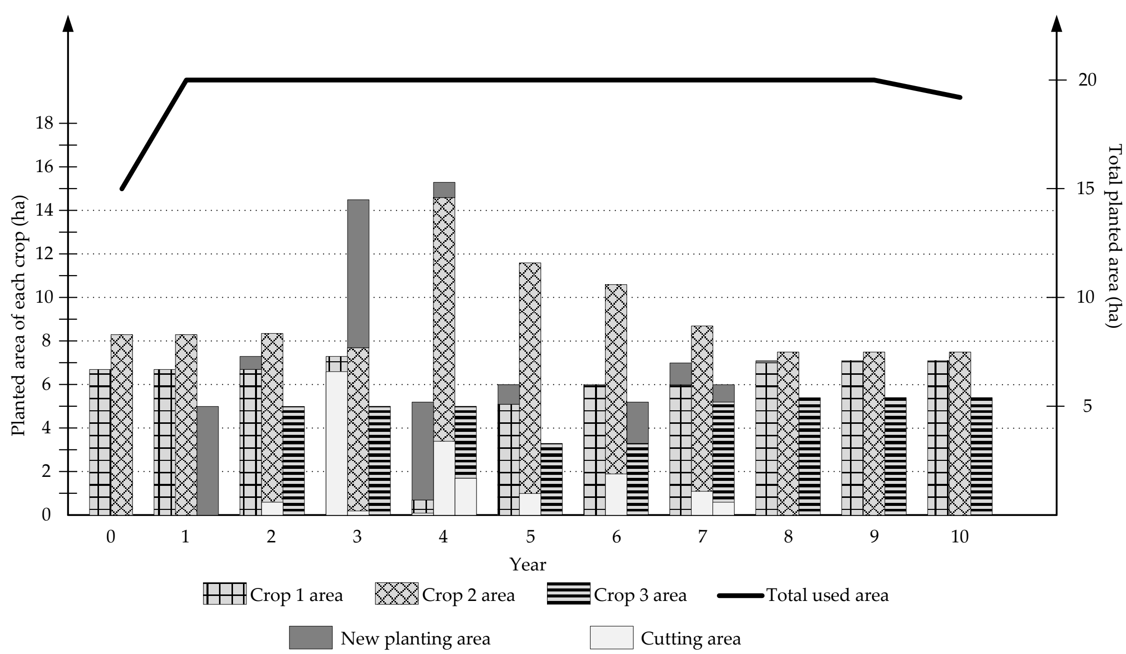

Figure 6.12 Plantation allocation when Crop 2 price decreases.

We can see effect the price of Crop 2 price in Figure 6.11 and 6.12. The total profits are also affected when the prices are changed (Figure 6.13).

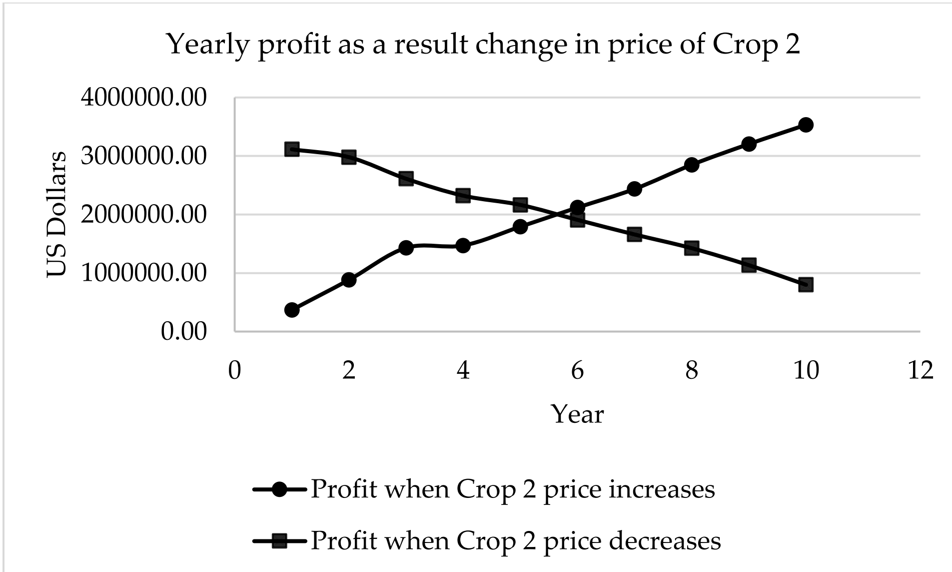

Figure 6.13 Effect of changes in Crop 2 price on profit.

6.5.3 Changing Demands

In this scenario, market demand variations are considered. It is assumed that the need of the market rises for all crops. The following cases are considered: Crop 3 demand increases four-fold and the demand for all crops required increases by 20%, 40%, and 80%, respectively, based on the baseline scenario demand. Other information is unchanged.

6.5.3.1 Crop 3 Demand Increases Four-Fold

In this case, it is supposed that the yellow-skin white-flesh dragon fruit is popular, so that the farmers have more orders from traders and wholesalers. The areas of new planting or cutting of the dragon fruit trees are shown below:

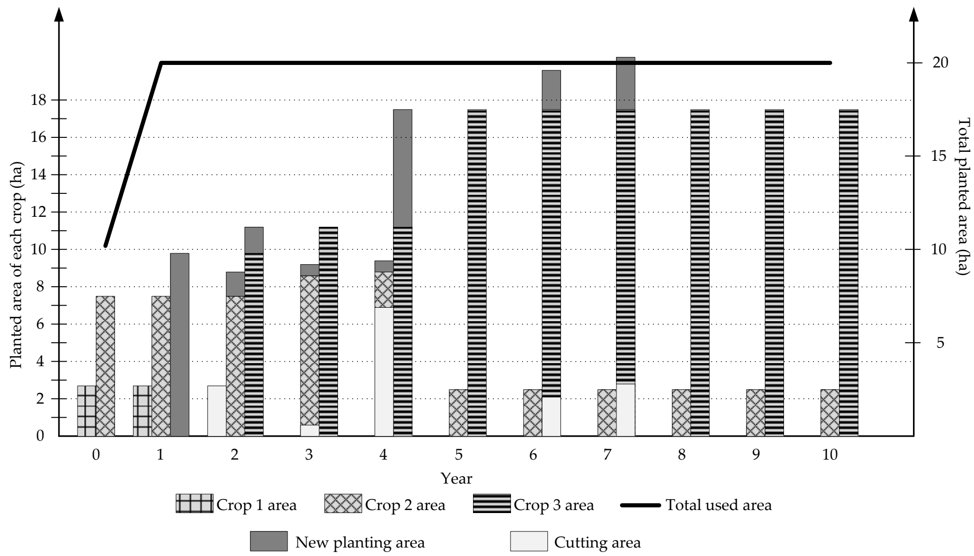

Figure 6.14 Plantation allocation when Crop 3 demand increases four-fold.

Figure 6.14 shows us that with the rising demand along with its high selling price, Crop 3 is produced on most of the current planting area. On the other hand, Crop 2 is grown on the small remaining land and Crop 1 is cut off completely in year 2.

6.5.3.2 All Crop Demands Increasing by a Fixed Percentage

This case is slightly different from the case "Crop 3 Demand Increases Four-Fold", and all crop demands increase by 20%, 40%, and 80% respectively over the baseline scenario demand. Other data are the same. The changes in plantation allocation for each case is shown in Figure 6.15, 6.16 and 6.17 below:

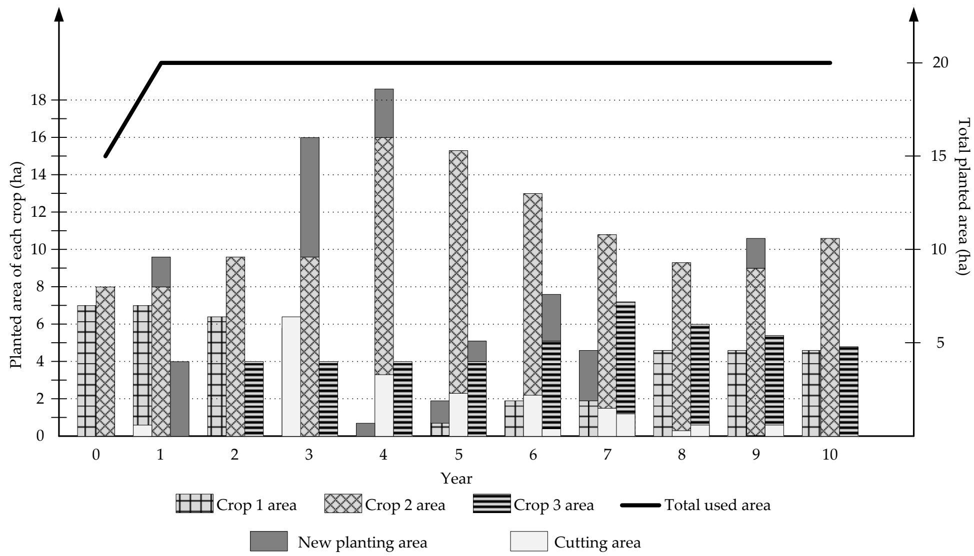

Figure 6.15 Plantation allocation when all crop demands increase by 20%.

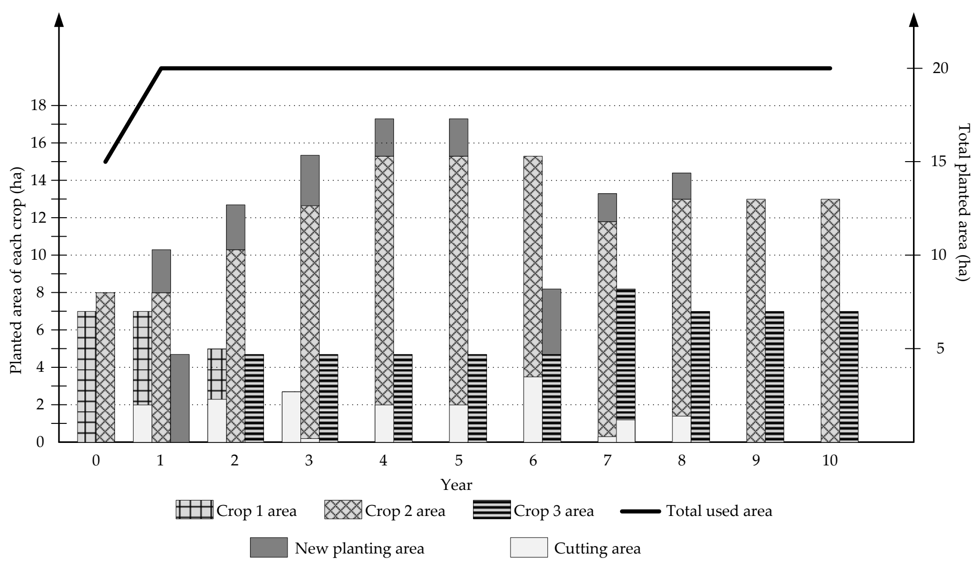

Figure 6.16 Plantation allocation when all crop demands increase by 40%.

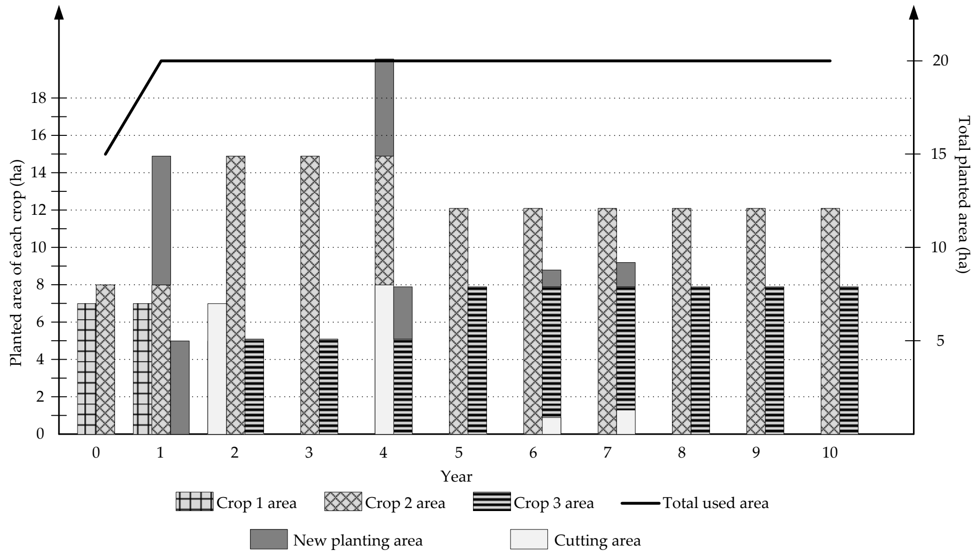

Figure 6.17 Plantation allocation when all crop demands increase by 80%.

We can see that if the needs of all crops are increased, only crops that are more profitable are grown. Therefore, Crop 2 and 3 are prioritized for planting. The profit of each case is summarized in Figure 6.18. In general, the higher the demand is, the more the dragon fruit grower's revenue is.

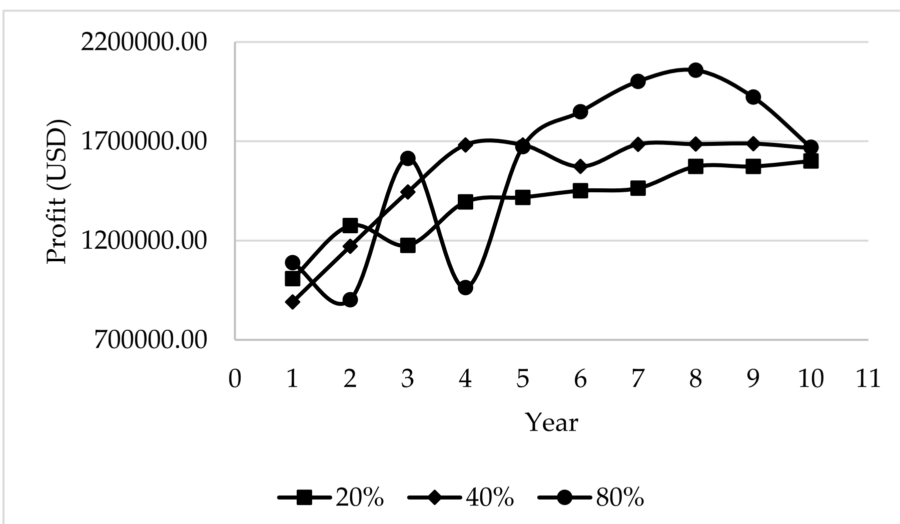

Figure 6.18 Profit for each case of increasing demand.

6.5.4 Crop 3 Selling Price with a Probability Factor

Currently, the bulk of Crop 3 is imported in the market. If it could be produced domestically, this demand would be less dependent on imports and could relieve price fluctuations. For this case, a probability factor is considered in the selling price. It is assumed three Crop 3 price scenarios of $1, $5, or $10 occur in each year of the planning period of 10 years. The probability for each price is assumed to be a discrete combination of 0.2, 0.2, and 0.6 for a total of 1.0 for the three price scenarios. The changes to the plantation for each case is presented below:

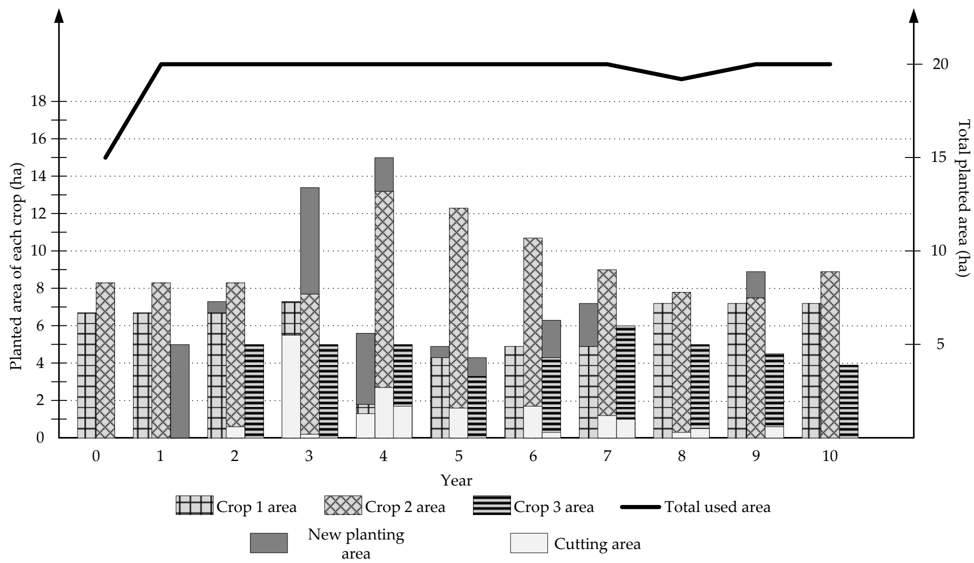

Figure 6.19 Plantation allocation with probability combination of prices$1 (0.2)–$5 (0.2)–$10 (0.6).

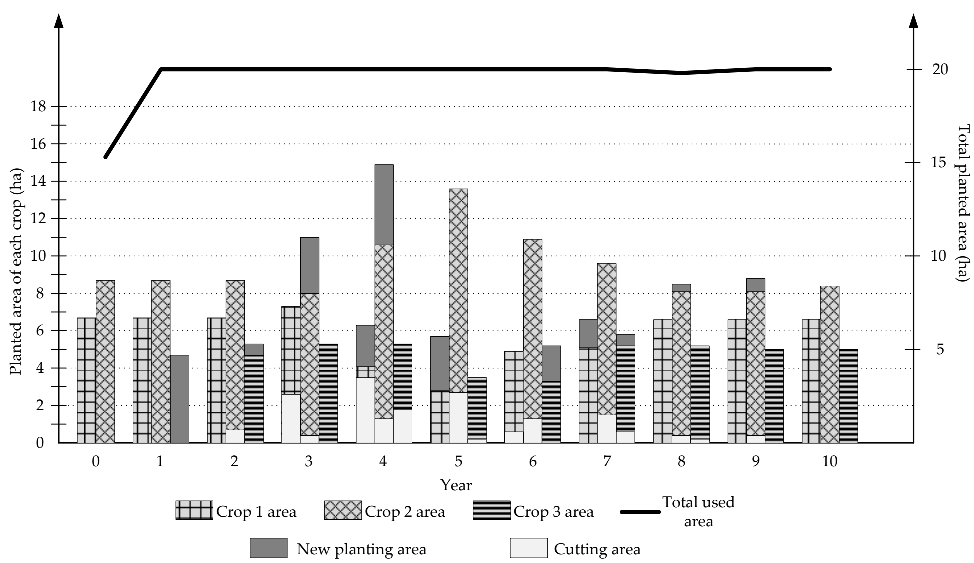

Figure 6.20 Plantation allocation with probability combination of prices $1 (0.2)–$5 (0.6)–$10 (0.2).

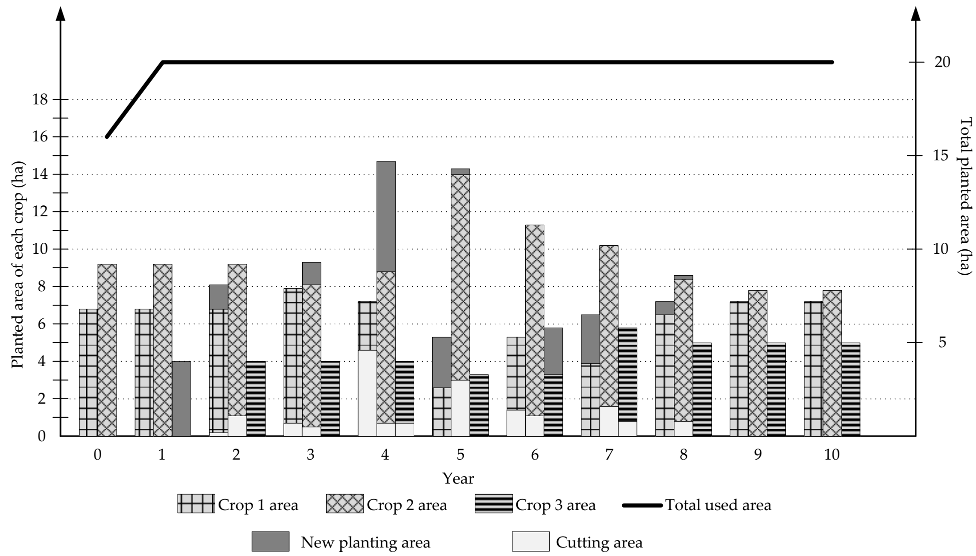

Figure 6.21 Plantation allocation with probability combination of prices \$1 (0.6)–\$5 (0.2)–\$10 (0.2).

Figures 6.19, 6.20, and 6.21 show that the higher the most probable price of Crop 3, the greater the land allocated to it in the first year. The area of Crop 3 for each case is 5 ha, 4.7 ha, and 4 ha, respectively. In addition, the change in area of cultivating land is also affected: the lower the most probable price, the higher the changeover to other crops. This can be observed in the 10-year profit projections (Figure 6.22).

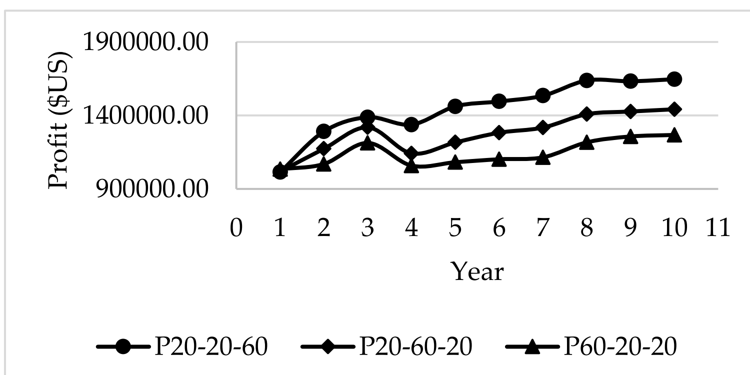

Figure 6.22 Profit for each case in scenario 6.5.4.

6.5.5 Land Restriction

This scenario was suggested by the farmers who want to not only have a stable plantation size for each crop, but also supply all three varieties of dragon fruits to the market. They would like to use 50%, 35%, and 15% of their land to produce Crop 1, Crop 2, and Crop 3, respectively, and this is in some sense a risk-hedging strategy. The cultivation activities (Figure 6.23) and the profit–cost relationship (Figure 6.24) for a 10-year horizon are shown below:

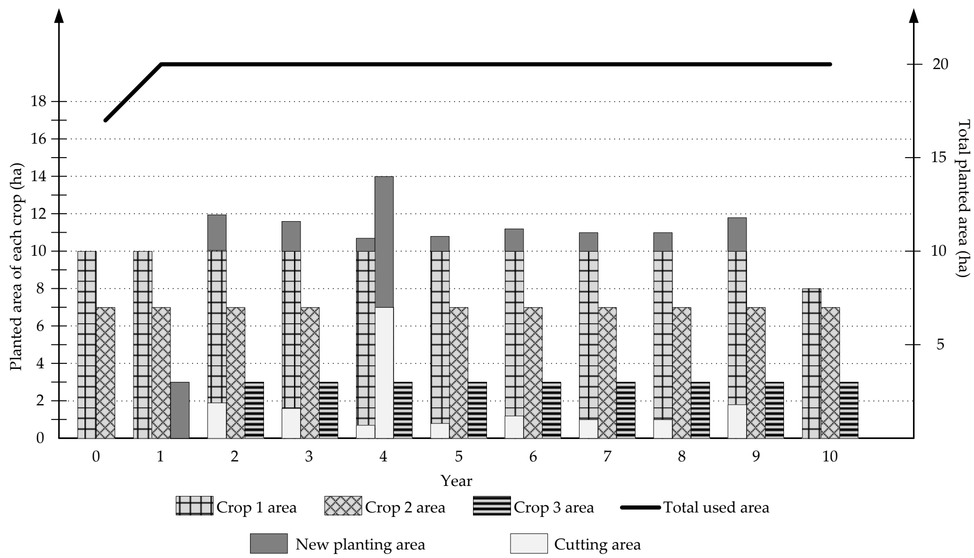

Figure 6.23 Plantation allocation for land restriction scenario.

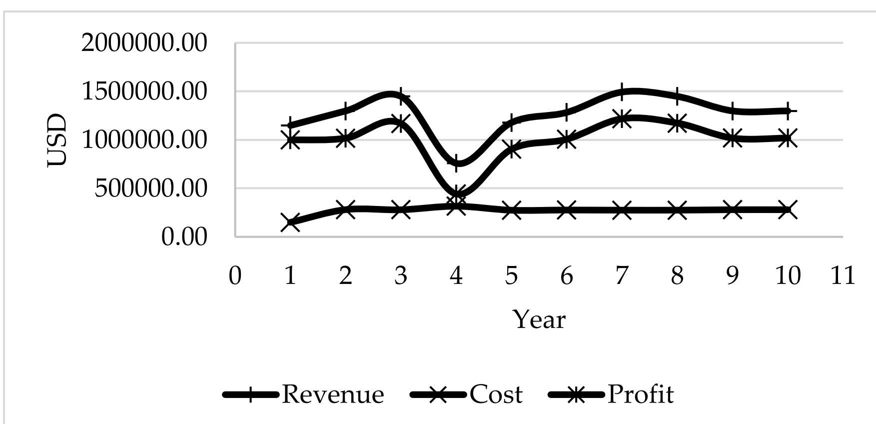

Figure 6.24 Revenue, profit, and cost in 10 years.

To ensure both the stability of the area for each crop and the amount supplied to the market, farmers should re-vitalize the trees every few years (cutting old plants and planting new ones), especially in the 4th year.

6.5.6 Influence of the Initial Plantation Conditions

In addition to the mentioned variables above, such as customer demand and selling prices, the initial crop status of the arable land is also considered to determine its influence on the changes in the planting area. Based on the result of the baseline scenario, two sub-scenarios are proposed to look at the effect of the variety of the initial crop and the age of the initial crop. In the first sub-scenario set, it is supposed that all initial area has been allocated to only one of the three crops (Crop 1 or Crop 2 or Crop 3). In the second sub-scenario set, only the initial land for Crop 1 is considered, with different ages (age 1, age 3, and age 5) in year 0. Other data is the same as in the baseline scenario.

6.5.6.1 Initial Plantation with Only One Kind of Crop

According to the results of the baseline scenario, the initial area that is used in year 0 is 15.3 hectares. It is assumed that only one crop has been planted on that area in year 0. Figures 6.25, 6.26, and 6.27 show the cultivation activities for each case.

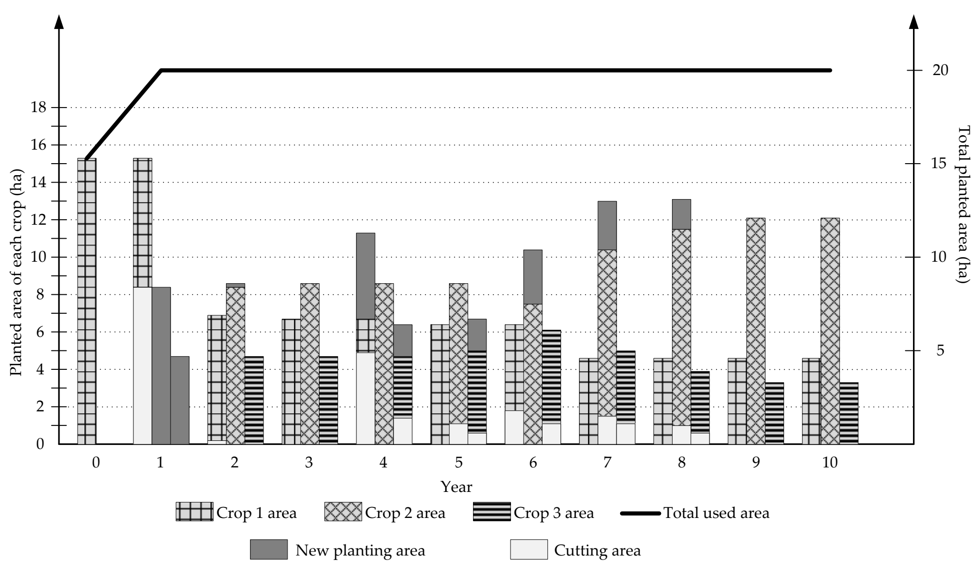

Figure 6.25 Plantation allocation if the initial plantation only has Crop 1.

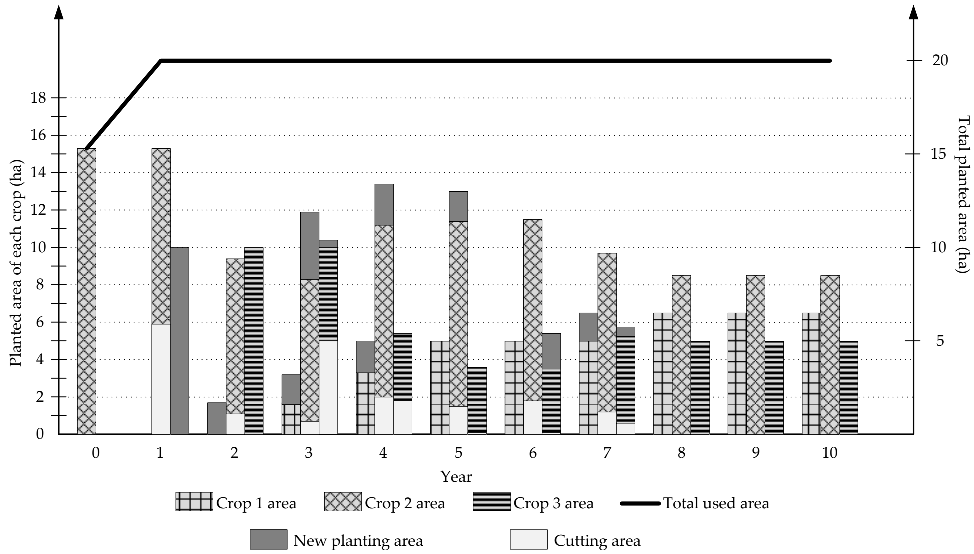

Figure 6.26 Plantation allocation if the initial plantation only has Crop 2.

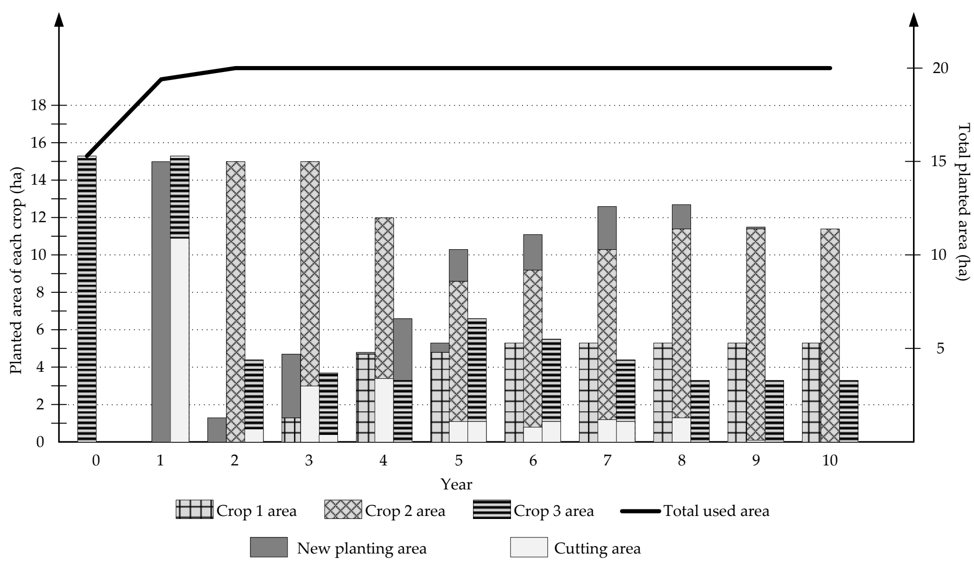

Figure 6.27 Plantation allocation if the initial plantation only has Crop 3.

Although the initial land varies across these sub scenarios, Crop 2 is still the most planted over the next 10 years. The profit for the case where the entire plantation consists of Crop 2 is also the highest, as shown in Figure 6.28 below:

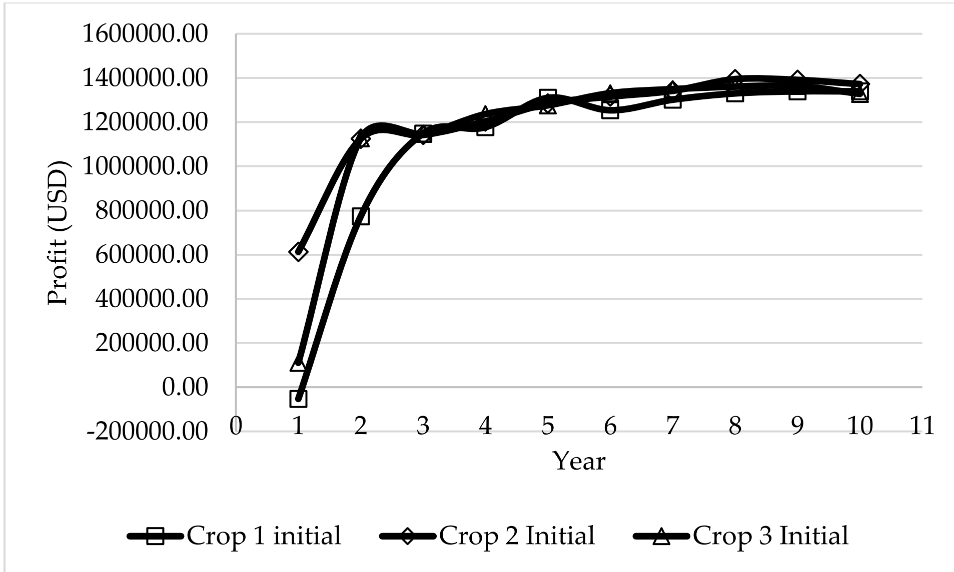

Figure 6.28 The profit of the three cases of the initial land allocation.

6.5.6.2 *The Initial Crop Allocation with Different Crop 1 Ages*

In this case, it is assumed that the initial crop allocation is only for Crop 1. However, it is assumed that the age is 1, 3, or 5 in year 0. The optimal crop allocation over the next 10 years is shown in Figure 6.29, 6.30, and 6.31.

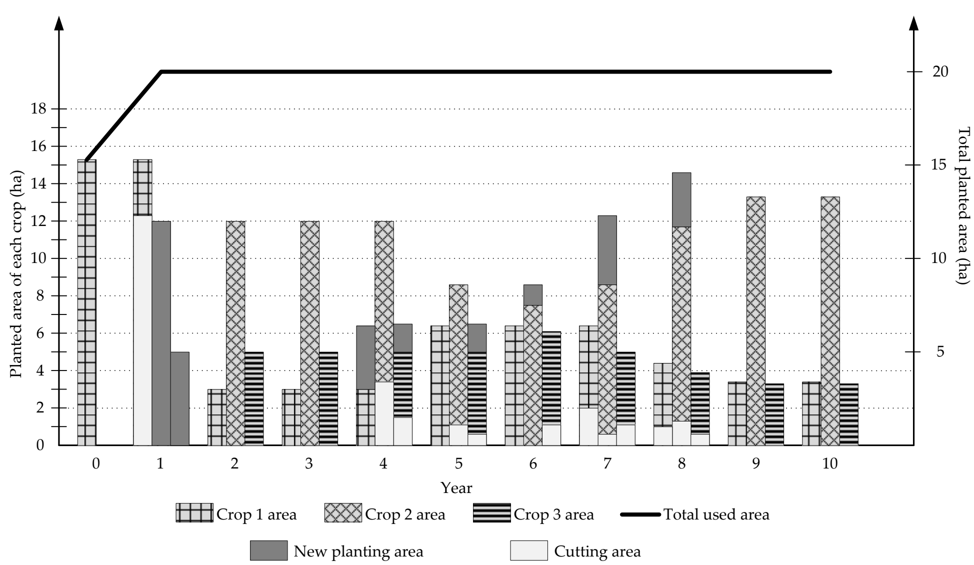

Figure 6.29 Plantation allocation if the initial land is only for Crop 1 at age 1.

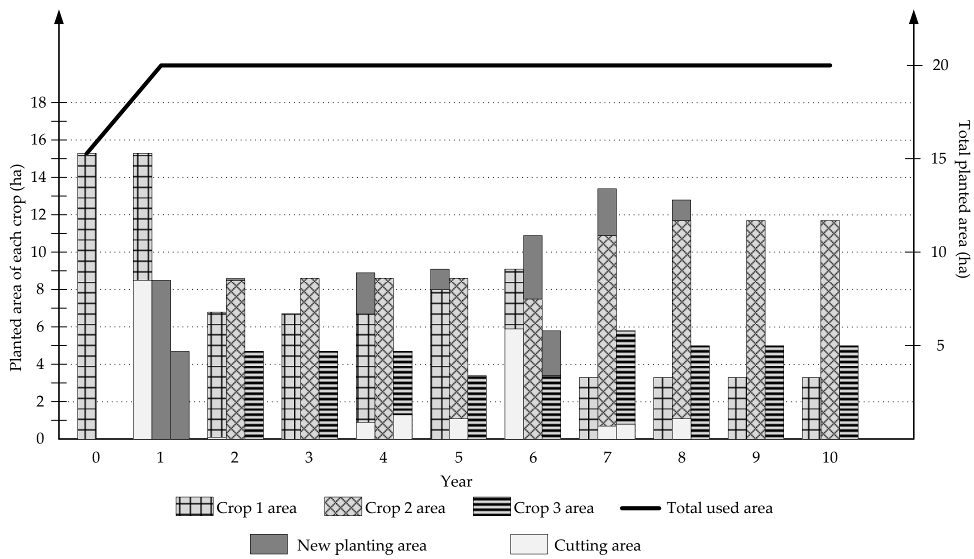

Figure 6.30 Plantation allocation if the initial land is only for Crop 1 at age 3.

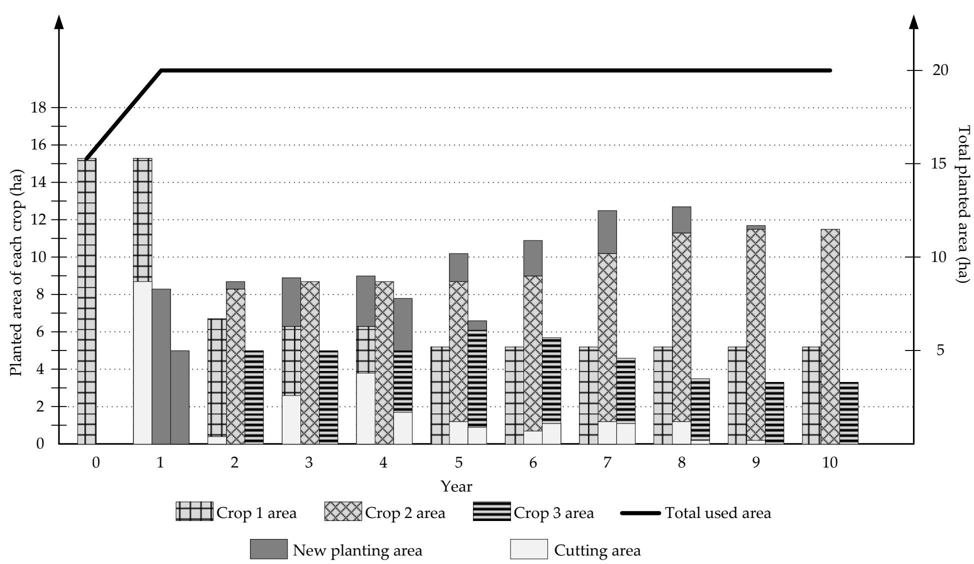

Figure 6.31 Plantation allocation if the initial land is only for Crop 1 at age 5.

We can see that the young Crop 1 (age 1) is truncated more than the grown one (age 3 or age 5). Similar to trends observed earlier, after Crop 1 area is reduced, most of the remaining land is prioritized for Crop 2, which has the most revenue because of its high demand. The profits for all three cases are shown in Figure 6.32.

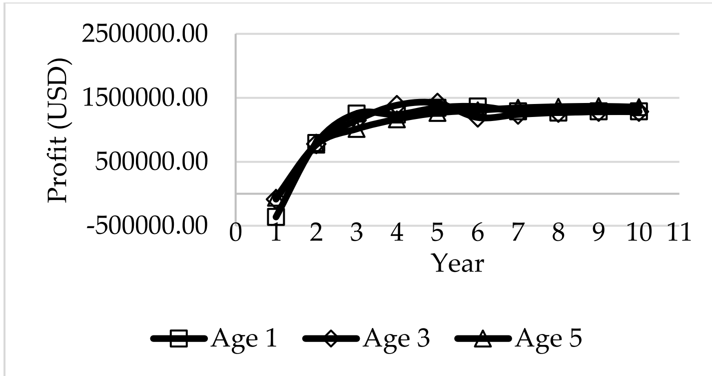

Figure 6.32 The profits for the three cases of initial land allocation to Crop 1 for different ages.

6.5.7 Discussion

Throughout scenarios and cases above, summarized in Table 6.2, we can see that the dragon fruit growers earn more profit if they prioritize planting the varieties for which both demand and selling price are high. This is the case for Crop 2 which has a very high demand and its price is only lower than the price of Crop 3. In contrast, although the price of Crop 3 is the highest, due to lower demand and yield, it is not prioritized in the solutions. Finally, the selling price of Crop 1 is the lowest; therefore, it is replaced by other varieties when the demands or prices of the other two crops increase. Table 6.2 summarizes the profits for each of the scenarios.

Table 6.2 Summary of the profitability of the various scenarios

Scenario	Sub-Scenario	Limit on Plantation Area for Each Crop	Profit
Baseline scenario		No limit for each crop	USD 12,576,086.80
Changes in price of Crop 2	1	No limit for each crop	USD 20,089,622.07
	2	No limit for each crop	USD 20,116,237.86
Changes in demands	1	No limit for each crop	USD 16,815,478.39
	2	No limit for each crop	USD 13,932,145.96
	3	No limit for each crop	USD 15,166,044.33
	4	No limit for each crop	USD 15,740,093.18
Crop 3 selling price with probability factor	1	No limit for each crop	USD 14,438,789.39

	2	No limit for each crop	USD 12,745,440.97
	3	No limit for each crop	USD 11,415,607.18
Land restriction		50% for Crop 1, 35% for Crop 2, 15% for Crop 3	USD 9,963,130.32
Influence of initial land	Crop 1	No limit for each crop	USD 10,921,697.80
	Crop 2	No limit for each crop	USD 12,182,762.60
	Crop 3	No limit for each crop	USD 11,630,711.50
	Crop 1—Age 1	No limit for each crop	USD 10,763,932.06
	Crop 1—Age 3	No limit for each crop	USD 10,907,367.04
	Crop 1—Age 5	No limit for each crop	USD 10,879,376.28

6.6 CONCLUSION

A deterministic model is proposed in this chapter to assist dragon fruit farmers with their decision making on crop allocation for different species of dragon fruits. Consequently, it can provide them a long-term overview through groups of production scenarios that could occur, such as 1) price changes (e.g., price of the red-skin red-flesh dragon fruit—Crop 2); 2) changes in demand (e.g., demand of the yellow-skin white-flesh dragon fruit—Crop 3); 3) requirements for land restrictions for each type of crop, and 4) the influence of the initial state. All scenarios are variants from a baseline scenario of the actual dragon fruit production conditions in Vietnam intended to provide insights. Results obtained from this model confirmed that Crop 2 should be prioritized for planting.

The model presented in this chapter represents a first step towards a comprehensive quantitative approach for decision making in dragon fruit cultivation in Vietnam. The model works well with some scenarios showing relationships of input factors (demands, prices, and costs) and output decisions (plantation area of crops) over a 10-year period. This is evident in the cases of "no limit plantation area" of all scenarios: the crop is grown if its demand and price increase, or it is cut down when its demand is low and its price drops. For a specific crop, the old plants are replaced by new ones if their yield is too low due to age.

The proposed model is a meaningful tool for managers and farmers to have a holistic view for long term planning with the goal of maximizing profits. It helps decide which varieties to plant proactively based on demand and price scenarios.

As with other fresh fruit supply chains, the dragon fruit chain faces challenges due to inherent uncertainties such as demands, price, and yield. This is the main limitation of the deterministic approach. Therefore, future research to deal with randomness and uncertainty for dragon fruit cultivation could involve stochastic programming [31, 32, 38] or robust optimization [123]. The approach can be generalized to other similar fresh fruit supply chains.

CHAPTER 7 STOCHASTIC MODELLING FRAMEWORKS FOR DRAGON FRUIT SUPPLY CHAINS IN VIETNAM WITH UNCERTAINTY

This chapter is based on the published article titled "Stochastic Modelling Frameworks for Dragon Fruit Supply Chains in Vietnam under Uncertain Factors" in *Sustainability*, 16(6), p. 2423 [124]. For more details, please refer to the electric copy that has been presented at https://www.mdpi.com/2071-1050/16/6/2423.

7.1 Abstract

Managing uncertainties and risks is always a difficult but fascinating task in fresh fruit supply chains, especially when dealing with the strategy for the production and conveyance of fresh fruit in Vietnam. Following the COVID-19 outbreak, the confluence of economic recession and persistent adverse weather conditions has exacerbated challenges faced by dragon fruit cultivators. This research investigates a two-stage stochastic programming (TSSP) approach which is developed and served as a valuable tool for analyzing uncertainties, optimizing operations, and managing risks in the fresh fruit industry, ultimately contributing to the sustainability and resilience of supply chains in the agricultural sector. A prototype is provided to illustrate the complex and dynamic nature of dragon fruit cultivation and consumption in Vietnam. Data on the selling prices of dragon fruit were collected from several sources between 2013 and 2022 in Binh Thuan Province, Vietnam. The results were obtained from the model by using three different approaches in order of their versatility and efficacy: (1) Scenario tree generation; (2) Sample average approximation; (3) Chance-constrained programming.

7.2 Introduction

Effectively managing a supply chain that can be impacted by volatile circumstances poses a significant challenge for all involved stakeholders. The supply chain of agricultural products, particularly fresh agricultural products, is inherently challenging due to its susceptibility to numerous unknown and unpredictable elements, such as climate change influences. A farm's longevity and production are adversely impacted by various factors, including drought, seawater intrusion, dangerous pests, illnesses, and overutilization of fertilizers and pesticides.

Uncertainty can manifest at several stages within the fresh fruit supply chain, encompassing production by farmers, post-harvest storage, processing, transportation, and distribution. Furthermore, it is worth noting that the level of uncertainty varies at each point and stage of the supply chain.

The year 2020 presented a multitude of arduous and demanding circumstances within the realm of agriculture, which have been unprecedented in nature for all individuals involved. The global outbreak of the coronavirus pandemic had a significant impact on farmers across several countries, while distinct meteorological conditions posed challenges to individual locations. The movement of agricultural products from production sites to end consumers shows significant challenges due to constraints on mobility, disruptions in supply chains, restrictions on borders or ports, escalating transportation expenses, and the closure of numerous markets.

The distribution of fruit and vegetable production is subject to significant fluctuations in both demand and prices. According to a report by the FAO in 2021, there has been a significant increase in the pricing of certain items, particularly those that are seen to have immune-boosting properties such as garlic, ginger, and all fruits rich in vitamin C. Conversely, the prices of other products have experienced a sharp decline [125].

The implementation of travel restrictions or limitations between nations exacerbates labor shortages, particularly in countries that heavily depend on seasonal labor [126]. Insufficient labor resources for timely harvesting may result in the spoilage of produce within the agricultural field. Delays in transit and unloading might result in damage to fresh produce while it is being stored in containers.

According to [127], there appears to be a shift in consumer purchasing patterns as a result of imposed limitations on travel. During the initial stages of the pandemic, there was a significant surge in consumer demand for stockpiling commodities, driven by dread and apprehension. Consequently, the fruit and vegetable market experienced a decline, with prices beginning to decrease. This can be attributed to customers purchasing larger quantities of merchandise, thereby contributing to the weakening of the market. Perishable fruits and vegetables saw reduced consumer demand, while their non-perishable counterparts, such as apples and carrots, exhibited greater purchasing power and were prone to price escalation.

7.2.1 Context of the Vietnamese Agriculture

The emergence of the COVID-19 epidemic exacerbated the existing challenges faced by Vietnam's agriculture sector, mostly manifested in decreased output levels and disruptions in agricultural supply chains. In rural regions, there has been a notable increase in the supply of agricultural commodities, including vegetables, flowers, fruits, and seafood, as a consequence of diminished consumer demand. Consequently, these excess commodities remain unutilized and, in certain instances, are subject to destruction. The oversupply has had a discernible influence on the market, resulting in a substantial decrease in prices, particularly for perishable agricultural commodities such as vegetables, flowers, fruits, and seafood. The disparity in pricing between farmers' selling prices and consumers' purchasing prices can be related to issues experienced within the circulation and distribution sector. The escalation of rice prices in the international market can be attributed to the surge in import demand from various nations. Consequently, this surge has resulted in a parallel increase in local prices, particularly for rice [128].

The data provided by The Ministry of Industry and Trade of Vietnam [129] shows that, in 2020, the export value of significant agricultural and fishery products had a decline compared to 2019. Specifically, the seafood industry generated a total revenue of USD 8.41 billion, experiencing a decline of 1.5%. Fruits and vegetables, on the other hand, reached a revenue of USD 3.27 billion, showing a significant decrease of 12.7%. Cashew nuts achieved a volume of 515 thousand tons, resulting in a turnover of USD 3.21 billion, which increased by 13.0% in volume but decreased by 2.3% in turnover. Coffee production amounted to 1.57 million tons, with a turnover of USD 2.74 billion, representing a decline of 5.6% in volume and 4.2% in turnover. Pepper production reached 285 thousand tons, generating a turnover of USD 661 million, which increased by 0.4% in volume but decreased by 7.5% in turnover. Lastly, tea production amounted to 135 thousand tons, with a turnover of USD 218 million, experiencing a decline of 1.8% in volume and 7.8% in turnover.

In the context of Vietnam, a nation mostly reliant on agriculture, it is noteworthy that the primary revenue stream for numerous farmers has also confronted adverse effects. From the commencement of 2020 till the present, the agricultural industry has experienced significant disruptions to its production and commercial operations, thereby impacting the financial well-being and livelihoods of farmers. The occurrence of disease poses significant challenges to various operations, including the provision of raw materials, trading, transportation, distribution, and exportation of agricultural products. Numerous enterprises, production facilities, and commercial establishments have experienced temporary closures or terminated contractual agreements, resulting in significant adverse effects on agricultural production. There is a significant prevalence of underemployment and unemployment among laborers in agricultural producing businesses, leading to a substantial decline in the average income of workers [130].

7.2.2 The Need for Mathematical Model of Fresh Fruit Supply Chains in the Vietnamese Context

The use of mathematical models provides substantial advantages in improving the production and distribution of fresh fruit. By tackling the obstacles and making well-informed decisions, these models may have a significant impact on establishing a fruit supply chain that is more effective, environmentally friendly, and adaptable. Nguyen et al. [131] stated that a wide range of mathematical models have been used in the last four decades to identify the most effective solutions for different requirements within the fresh fruit supply chain. Previous review papers [47-51] on mathematical models applied to the agri-food supply chain demonstrate that stochastic models are capable of efficiently addressing challenges including risk and uncertainty.

This chapter, in consideration of all above-mentioned contexts, aims to examine the uncertainties associated with the production and distribution of fresh fruit in Vietnam. It specifically focuses on the cultivation of dragon fruit as a case study and the use of a two-state stochastic programming model was involved to respond to the need for sustainable solutions under uncertainties. The deterministic model developed by Nguyen et al. [41] served as a valuable tool for analyzing uncertainties, optimizing operations, and managing risks in fresh fruit production and distribution. Ultimately, this model has contributed to the sustainability and resilience of supply chains in the agricultural sector. The results were obtained from the model by using three different approaches in order of their versatility and efficacy: (1) Scenario tree generation; (2) Sample average approximation; (3) Chance-constrained programming. The application of these methods to the two-stage stochastic model is a potential new direction to address the uncertainties that affect the production and distribution of fresh fruit. Based on the solutions derived, a comparison will be made to assist managers or decision-makers in determining the most suitable strategy for their needs.

But, before presenting our conceptual framework and model, we try to describe a panoramic picture of dragon fruit in the Vietnamese market, to review some contexts of mathematical models existing in the literature for this issue.

7.3 Context of Dragon Fruit in Vietnam: Emerging Trends of Modelling Frameworks and Our Conception

7.3.1 The Impact of Uncertain Factors on Vietnam's Dragon Fruit Industry

Price factor: Due to China's status as the primary market for Vietnamese agricultural products, the onset of the COVID-19 epidemic in China in January posed significant challenges for the procurement and exportation of agricultural goods, including dragon fruits, to the border. Shipments destined for the border are currently experiencing congestion due to insufficient customs processing procedures, compounded by the suspension of sea exports. Prior to the commencement of the Lunar New Year, the prevailing market rate for white dragon fruit exceeded USD 1 per kilogram. However, subsequent to the Lunar New Year, the price experienced a significant decrease, plummeting to a mere USD 0.1 per kilogram. Consequently, no traders have shown interest in procuring the aforementioned commodity [132, 133].

Furthermore, the domestic selling price of dragon fruit has a paradoxical nature. An illustrative instance can be observed in the case of dragon fruit with yellow skin, which represents one of the three primary varieties of dragon fruit available in the Vietnamese market. In 2004, the Southern Horticultural Research Institute (SOFRI), Vietnam conducted an experiment including the importation and cultivation of Hylocereus costaricensis, a kind of dragon fruit characterized by its yellow peel and white flesh. However, the outcomes did not align with the anticipated expectations. The growth of the plants was suboptimal, characterized by thin and slender branches, as well as the production of little fruits weighing less than 200 g of each. Furthermore, the yield was found to be low, with an average of 2–3 kg per pole every season for 3-year-old plants. In order to satisfy the significant domestic demand and cater to the curiosity of consumers, the fruit is imported from Malaysia and retailed at a price that is up to 20 times greater than that of the red skin and white meat variants [134].

Given the perceived market potential of this novel fruit, numerous horticulturists engage in self-propagation and undertake planting endeavors, commencing in 2018 [135]. Nevertheless, because of the lack of prior knowledge in tree care, the trees exhibit a phenomenon wherein they produce flowers but fail to yield any fruit. Alternatively, there exist fruits that possess unattractive physical characteristics, rendering them unsuitable for commercial transactions. According to a dragon fruit garden owner, there was a significant demand for this fruit in the past due to its high price. As a result, farmers expressed interest in cultivating it. However, the unsuitability of the climate, soil, and cultivation practices for this variety, as opposed to the red skin dragon fruit, hindered their ability to achieve the intended outcomes. Growers face the challenge of declining pricing when they possess knowledge of the methods for cultivating yellow-skin dragon fruits, with prices often ranging from equal to or twice that of red meat dragon fruits, provided the fruit exhibits substantial size and an aesthetically pleasing appearance.

Furthermore, the exportation of the Vietnamese yellow-skin dragon fruit remains unfeasible. The underlying cause is that major consumer markets for dragon fruit from Vietnam, such as China, the European Union, and North America, exhibit a lack of preference for this product. China, being the primary consumer of dragon fruit from Vietnam, exhibits a notable preference for red dragon fruits, whether in terms of their skin or flesh.

Impact of climate change: Climate change can potentially affect multiple facets of crop water demand, crop growth and production, irrigation water supply, as well as the occurrence of floods, droughts, and heat waves.

The Global Climate Risk Index 2020 indicates Vietnam to be the sixth nation globally in terms of its exposure to climate change and extreme weather events throughout the period spanning from 1999 to 2018. It can be foreseen that climate change will lead to a more frequent occurrence of natural disasters and extreme heat waves in the majority of Vietnam [136].

According to a report by the Asian Development Bank [137], the risks associated with climate change have been found to impact various socioeconomic aspects in Vietnam including water management, pricing, allocation, access to finance, labor cost, and market price.

Binh Thuan Province is situated in the southeastern region of Vietnam, characterized by an extensive coastline and a monsoon-influenced tropical climate, resulting in well-defined wet and dry seasons. The wet season is typically observed between the months of May and October, whilst the dry season spans from November to April. The region has experienced a range of impacts as a result of climate change, including elevated sea levels, heightened temperatures, and alterations in precipitation patterns.

Binh Thuan, being situated in a coastal area, is susceptible to the consequences of escalating sea levels, encompassing coastal erosion, inundation, and the infiltration of saline water into freshwater reservoirs. The available data indicates that there has been a consistent annual increase in the sea level of approximately 3 mm during the period from 1993 to 2008. Projections suggest that, by the year 2050, the sea level is expected to climb within the range of 28 cm to 33 cm [138].

The region, like other parts of the globe, has encountered a rise in average temperatures under the effect of climate change. The potential consequences of this phenomenon extend to various aspects, including ecology, agriculture, and human health. The mean temperature had a gradual increase ranging from 0.5 to 0.7 degrees Celsius during the period from 1958 to 2007. It is projected that the average temperature in 2050 will experience a further increase of 0.4 degrees Celsius compared to the average temperature observed in 2020 [138].

The occurrence of soil erosion, desertification, and drought in Binh Thuan can be attributed to a confluence of factors, including rising temperatures, an upsurge in the frequency of sunny days, and intensified hot winds originating from the mainland during the dry season [139].

The average annual temperature data for Phan Thiet, Binh Thuan, as recorded and compiled by Meteoblue [140] from 1979 to 2021 (Figure 7.1), indicates a discernible trend towards increasing temperatures, with consistently higher values observed since 2010. In addition, the years 1998, 2016, 2019, and 2020 exhibited the highest average temperatures, accompanied by the occurrence of severe and protracted drought conditions [139, 141].

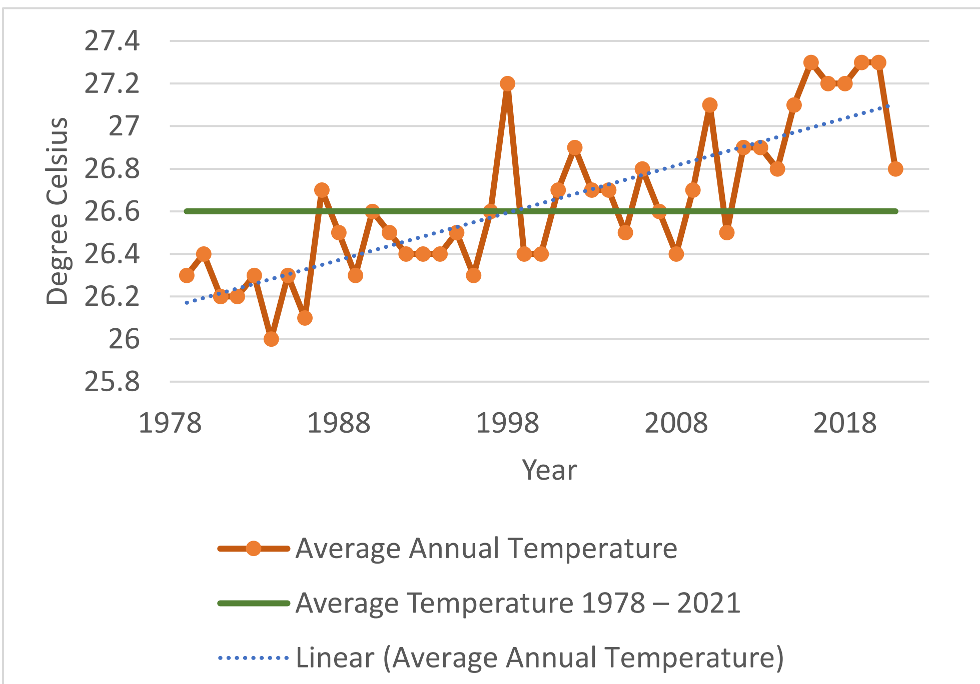

Figure 7.1 Trendline of yearly temperature in Binh Thuan (1979–2021).

The phenomenon of climate change has caused modifications in precipitation patterns, hence giving rise to heightened occurrences of severe weather events including intense rainfall, storms, and periods of drought. Based on the analysis conducted by Doutreloup et al. [142], it is anticipated that the duration of the dry season will be extended due to the projected shift at the beginning and conclusion of the wet season within the time frame of

2046–2065. While there may be alterations in the seasonal patterns of rainfall, it is expected that the overall annual precipitation will remain constant. Similar findings can be drawn for the period spanning from 2081 to 2100. Consequently, the climatic conditions in Binh Thuan Province undergo alterations characterized by an extended period of aridity, intensified summer precipitation, and heightened occurrences of heavy rainfall.

In recent years, the south–central region has experienced significant rainfall fluctuations attributed to the influence of the El Niño–Southern Oscillation (ENSO) phenomenon. This phenomenon, characterized by the simultaneous occurrence of El Niño and La Niña, has resulted in an increased variability in rainfall patterns. Specifically, the region has witnessed a higher frequency of years with below-average precipitation, leading to a substantial reduction in total annual rainfall. In some instances, the total annual rainfall has been observed to be more than 20% lower than the long-term average, with certain years experiencing reductions exceeding 30% [143]. In additionally, according to research by Vinh and Huong [144] combined with forecasting models proposed by the Ministry of Natural Resources and Environment of Vietnam [145], drought and water shortage will become more and more serious; specifically, by 2050, there will be no shortage of water. There is still a significant drought area; by 2100, there will be only severe drought in Binh Thuan. Figure 7.2 below shows a decreasing trend in average annual rainfall between 1979 and 2021 [140].

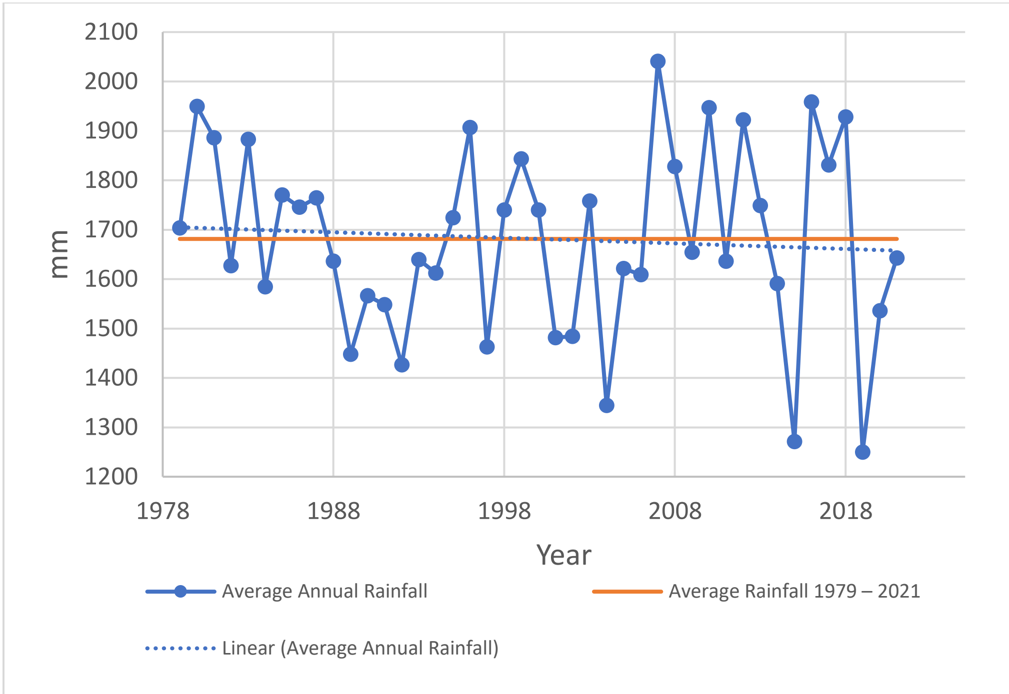

Figure 7.2 Trendline of yearly rainfall in Binh Thuan (1979–2021).

Despite the diminishing duration of the rainy season, the heightened occurrence and magnitude of intense precipitation events can give rise to flash floods and landslides, presenting a significant risk to human lives and assets. Furthermore, an abundance of precipitation has the potential to cause harm to infrastructure and have adverse effects on agricultural productivity. The occurrence of these events can exert substantial impacts on the agricultural sector, water supplies, and infrastructure within the region [146].

The agricultural sector in Binh Thuan Province is susceptible to the impacts of climate change, particularly in relation to crop production. Changes in temperature and precipitation patterns can have detrimental effects on agricultural productivity. Extended periods of drought and severe weather phenomena have the potential to diminish agricultural yields and alter established farming methodologies.

The agricultural sector in Binh Thuan Province is likely to see substantial effects as a result of increasing temperatures. According to the World Economic Forum, the adverse effects of climate change, such as elevated temperatures and intensified precipitation patterns, are causing detrimental impacts on land quality and leading to a decline in soil production. The negative impact on crop production is a consequence of the depletion of organic matter and soil nutrients [147]. Moreover, it is worth noting that alterations in temperature, atmospheric carbon dioxide (CO_2) levels, and the occurrence and severity of extreme weather events may exert substantial effects on agricultural productivity [148].

The occurrence of pests and illnesses that have the potential to impact fruit output is being influenced by climate change. For example, alterations in temperature and precipitation patterns have the potential to establish conducive environments for the proliferation and dissemination of pests and diseases, which can inflict harm upon fruit trees, fruits, and foliage.

In summary, the phenomenon of climate change is exerting various effects on the production of fresh fruit in Binh Thuan. These effects encompass alterations in temperature and precipitation patterns, the incidence of extreme weather events, as well as shifts in the prevalence of pests and illnesses. The aforementioned repercussions have the potential to lead to decreased agricultural production, diminished fruit quality, and financial losses for both farmers and the surrounding community.

7.3.2 Emerging Trends of Modelling Frameworks for Uncertainty and Climate Effects

Managers have been utilizing and developing decision support systems in response to the uncertainties associated with fresh fruit production/distribution. The persistence of uncertainty and the challenges associated with forecasting continue to be a matter of concern [149].

The agri-food supply chain is a multifaceted system encompassing production, distribution, and consumption. It is marked by considerable volatility arising from various causes, including weather conditions, market fluctuations, and customer preferences. Researchers have additionally presented a diverse range of mathematical models aimed at assisting managers and direct farmers in mitigating errors in decision-making within the context of risks and uncertainties prevalent in the agricultural supply chain. Nguyen et al. [39], who aim to improve the supply chains of agricultural goods, are the main researchers that use the deterministic model. The decisions made by managers, however, may be prone to error due to the inherent nature and limitations of the deterministic model, which lacks the ability to effectively address uncertainties and risks. In recent years, there has been an emergence of stochastic programming and resilient optimization models as viable approaches to address problems characterized by uncertain aspects. The aforementioned methodologies have undergone enhancements, rendering them valuable instruments for decision-makers to promptly and efficiently tackle challenges pertaining to manufacturing, processing, transportation, and distribution.

Stochastic linear programming (SLP) is an extension of linear programming that incorporates parameters with inherent uncertainty. It has been applied in agri-food supply chain management, particularly in crop planning and animal production. Researchers like Carøe and Schultz [150] have developed dual decomposition, a method for breaking down large stochastic integer programs into smaller subproblems. Pourmohammadi et al. [151] created a model considering production, transportation, storage, processing, and regional demand to optimize wheat supply chains under uncertainty. The model reduced costs and improved supply chain performance. Jacquet and Pluvinage [152] developed a discrete stochastic programming model to investigate how climatic variability affects farm management and evaluate farm strategies under different weather conditions. The model optimizes farm income and evaluates agricultural strategies in the setting of climate uncertainties. The study concluded that diversification and insurance policies may help farmers handle climatic uncertainty risks, emphasizing the importance of climate unpredictability in agriculture policy design and decision-making.

Two-stage stochastic programming (TSSP) is a widely used method in supply chain management, particularly in the agri-food industry. It helps manage the trade-off between long-term and short-term decision-making, considering uncertainty. Studies by Darby-Dowman et al. [31], Kazaz [32], Ahumada et al. [33], Tan and Çömden [153], Costa et al. [154], Marchal et al. [155], and Flores and Villalobos [156] have all highlighted the importance of TSSP for managing uncertainty in supply chain management.

Darby-Dowman et al. [31] developed a TSSP model with recourse model for horticulture planting plans, which accounts for weather and crop output variables. Kazaz's production planning model [32] considers yield, demand, cost, and price interdependencies to optimize production decisions. Ahumada et al. [33] optimized production and distribution using a two-stage stochastic mixed-integer linear programming approach (SMILP), accounting for demand fluctuations, yield variability, and transportation costs. In their work, Tan and Çömden [153] optimized yearly crop planning by addressing multiple sources of uncertainty, including variations in demand, maturation, harvest, and yield hazards. Costa et al. [154] provided a paradigm for managing perishable vegetable crop supply chains, considering agricultural production, transportation, storage, and product perishability. Marchal et al. [155] created a SMILP model to optimize production planning and address uncertainties. Flores and Villalobos [156] suggested a stochastic planning framework to help agricultural stakeholders integrate different systems while considering market prices, crop yields, and weather variables. The approach optimizes land allocation, production, and resource management, maximizing predicted profit while considering land, labor, and environmental constraints.

Robust optimization (RO) is a method that considers multiple scenarios to create less uncertain solutions. Bohle et al. [29], Munhoz and Morabito [30], and An and Ouyang [157] have used RO in agri-food supply chain management to improve resilience and reduce uncertainty-related risks.

A model developed by Bohle et al. [29] optimizes harvest timing to maximize grape quality and minimize labor and equipment expenditures, improving wine grape harvesting schedule efficiency and dependability. Munhoz and Morabito [30] created a robust citrus firm production planning optimization model using recourse actions and two-stage SMILP. The model minimizes the total estimated cost of the production plan, including supply and demand uncertainties, and meets customer demand and product quality standards in uncertain settings. In 2016, An and Ouyang [157] introduced a resilient grain supply chain design model that accounts for post-harvest loss and harvest timing equilibrium, resolving supply–demand uncertainty and harvest timing–crop yield trade-offs. The model accurately represents the complicated harvest timing and post-harvest loss trade-offs, resulting in more robust and efficient supply chain architectures.

Multi-objective Stochastic Programming (MOSP) models have been utilized to manage the agri-food supply chain, focusing on economic, environmental, and social trade-offs. Banasik et al. [158] developed a decision-support tool that considers demand, production yield uncertainty, and environmental factors. They used a mixed-integer linear programming (MILP) methodology to reduce production costs and environmental effects. The model allows producers to adjust their schedule based on demand and yield, resulting in more robust and eco-efficient production plans. Chavez et al. [159] suggested a multi-objective stochastic optimization model for scheduling upstream operations in a sustainable sugarcane chain while considering growth, harvest, transportation, manpower, machinery, and vehicle scheduling. A compromise programming methodology was used to find the best harvesting method trade-offs in a Peruvian case study.

Global warming threatens natural resources, ecosystems, and human society [160]. Climate change is worsening the impact of weather patterns on crop yields, productivity, and quality. Climate change significantly impacts agriculture, including fresh fruit production, due to temperature, rainfall, and extreme weather events [161, 162].

Duangdai and Likasiri [163] used mathematical modelling to study the relationship between global temperature and forest coverage, with a significant negative association found between temperature and rainfall. Lim et al. [106] proposed a two-stage optimization model to increase oil palm plantation harvesting and transport efficiency, which reduced journey distance and improved harvesting and transportation efficiency. These findings can help plantation managers allocate resources more efficiently, increasing production and lowering costs. Sun et al. [164] studied climate change and vegetation patterns using mathematical modelling and data analysis, highlighting the need for understanding climate change's complex interactions with vegetation dynamics to develop effective conservation and management strategies. Ghaffari et al. [165] used a Positive Mathematical Planning (PMP) model to evaluate drought's economic consequences on agriculture under various climate change scenarios, emphasizing the need for effective adaptation and mitigation strategies. Kung and Wu [166] examined how climate change influences water allocation and bioenergy output using stochastic mathematical programming. They found that water availability significantly impacts bioenergy output and that water distribution strategies may have different effects. Climate change management requires efficient water distribution, and adapting to water availability, crop yield, and unpredictability is crucial. This research highlights the importance of managing climate change sustainably to ensure sustainable development of water resources, agricultural productivity, and bioenergy.

As a matter of fact, studies on stochastic programming models for the agri-food supply chain need to include the following:

- Enhanced prediction and decision-making through machine learning and artificial intelligence integration.
- The development of two-stage stochastic programming (TSSP) models to enhance supply chain management.
- The expansion of stochastic models to address climate change, sustainability, and circular economy.

7.3.3 Approach and Model

To address the intricacies of decision-making processes and the prevalence of uncertain inputs, several mathematical and statistical methodologies are often explored. Among them, nonlinear programming and stochastic programming are particularly prominent. This study introduces and applies stochastic programming and robust optimization techniques, including the scenario tree method, sample average approximation, and chance-constrained formulation. These approaches are utilized to construct a mathematical model that addresses the uncertainty associated with selling prices within the context of the real food system.

The stochastic model developed for a normal growing season can consist of two distinct phases.

The initial stage encompasses strategic choices that are exclusively made at the commencement of the agricultural season, including the selection of crops, determination of optimal planting quantities, and establishment of appropriate cultivation timelines.

During the second phase, farmers make adjustments to the decisions that were previously made in the first phase as the season develops. The farmers are faced with the task of determining the optimal quantity to harvest throughout each season and making decisions regarding which customers to sell their produce to, based on prevailing market conditions. The model additionally incorporates the probabilistic characteristics of the dragon fruit production and distribution processes.

The subsequent phase involves the consideration and analysis of uncertain factors that exhibit either high risks or low probability but large magnitude risks. The variability in risk levels associated with different uncertainties is contingent upon the specific cultivar of dragon fruit, as stated in Table 7.1.

Table 7.1 Risk level of uncertain factors for each variety of dragon fruit

	Variety of Dragon Fruit		
Factor	Red skin	Red flesh	Yellow skin
Weather conditions	Moderate	Moderate	Moderate
Yield	Moderate	Moderate	High
Market price	Low	Moderate	High
Demand	Moderate	Moderate	High

7.4 METHODOLOGY

7.4.1 Conceptual Framework

The proposed models for research on the fresh fruit supply chain in Vietnam are developed based on existing problems of production and processing dragon fruit to deal with the uncertain parameters of the agriculture by using stochastic optimization and robust approaches. The models handle middle- and long-term decisions such as crop planting plans, growing and harvesting plans, and distribution plans for dragon fruit plants in Vietnam. A basic network of dragon fruit production and processing is considered and described in Figure 7.3 that presents the different combination of transactions between parties of the supply chain network.

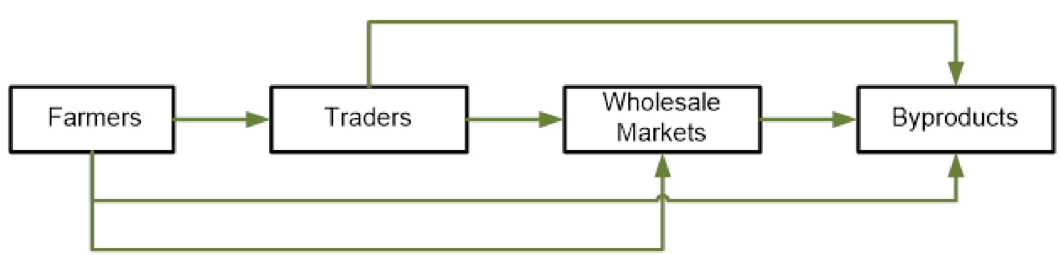

Figure 7.3 Dragon fruit production and processing chain.

Firstly, a recourse stochastic model is extended from the deterministic model proposed by Nguyen et al. [41] and it includes two stages. To make decisions that are reliant on random variables such as the market price, the TSSP approach is suggested and summarized in

Figure 7.4. Due to the characteristics of perennial tropical fruit plants, the first-stage solution for the problem is made considering planting factors such as constraints of land, water, labor, limitations on investment for first years of planting, and annual costs for year-by-year crops. The objective of the first stage is to support the farmers to evaluate and make better decisions and policies that not only increase the expected revenue but also reduce their risk by considering different scenarios.

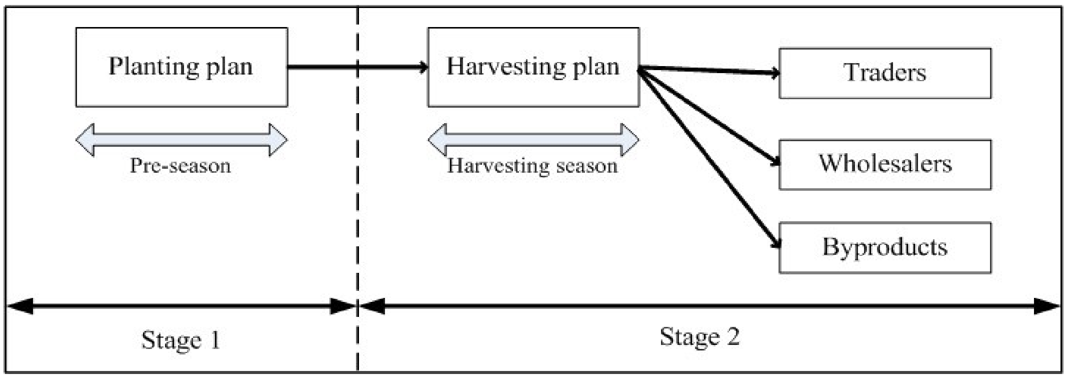

Figure 7.4 The two-stage decision-making process for fresh fruit production and distribution.

In the second stage, to provide satisfactory service to customers, the cognitive process to make decisions requires trustworthy and detailed statistics such as the capacity to harvest and the quantity to ship to selected customers while the prices fluctuate. Then, to better provide detailed distribution plans, reduced production and market uncertainty are approximated using the vicinity of the operational planning to the actual harvesting period. The environment of decision-making of this TSSP model is described in Figure 7.5.

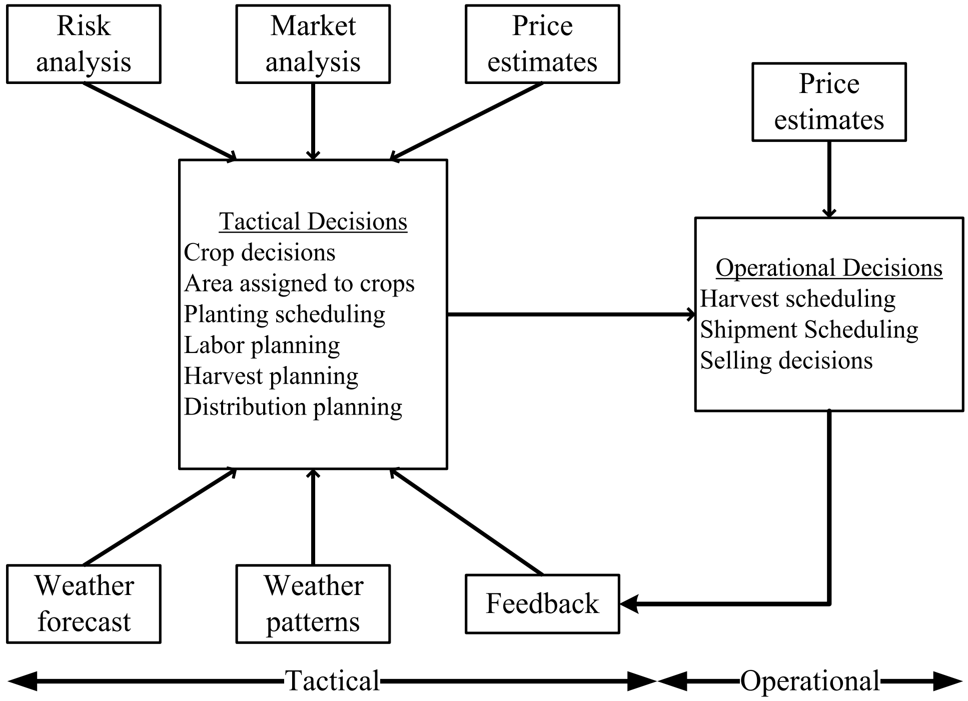

Figure 7.5 The environment of decision-making of the recourse two-stage stochastic model (TSSP).

This study presents two sampling procedures that are proposed as appropriate approaches to tackle the uncertain difficulties associated with fresh fruit production and delivery in Vietnam. The approaches employed in this study include scenario tree generation and sample average approximation. These methodologies are designed to conceptualize and evaluate ambiguity in order to identify feasible resolutions to the issue. The inherent flexibility and scalability of both approaches [38] offer significant advantages, making them effective instruments for addressing difficulties in stochastic programming. Figure 7.6 shows the methodological steps that are implemented in this chapter.

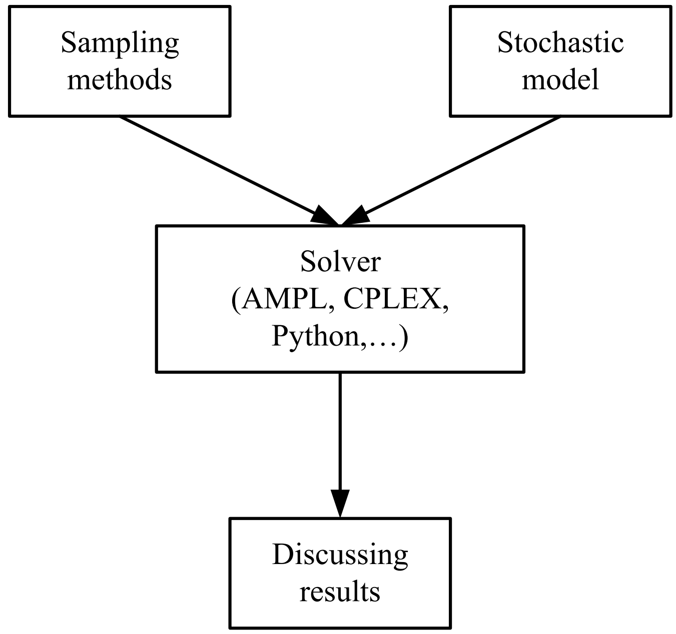

Figure 7.6 Methodological procedures implemented for the study of the production and distribution of dragon fruit in Vietnam.

The model was tested by using a dataset of a Vietnamese fresh fruit to verify the validity of results. The data were collected using methods such as surveys and direct observation. In the modelling phase, the historical data taken from the Department of Agriculture and Rural Development of Binh Thuan Province and General Statistics Office of Vietnam were used to validate the results. To ensure the usefulness and the applicability of the models in real life, the data used to validate were collected directly from farmers, traders, and wholesalers who are involved in a fresh fruit supply chain by using interviewing or performing surveys.

7.4.2 Formulation of the Stochastic Model

- Objective: maximizing the profit of crop harvesting for T years.

$$\max \sum_{s} \rho_s \Big(\sum_{j}\sum_{i}\sum_{t} p_{jits}\, ST_{jits} + \sum_{j}\sum_{m}\sum_{t} q_{jmts}\, SWM_{jmts} + \sum_{j}\sum_{t} r_{jts}\, SB_{jts}$$
$$- \sum_{j}\sum_{i}\sum_{t} c\,PNT1_{jit}\epsilon^{1}_{jits} - \sum_{j}\sum_{m}\sum_{t} c\,PNT2_{jmt}\epsilon^{1}_{jmts} - \sum_{t}\left(cbp_t \sum_{j} S\,B_{jts} \right)\Big)$$
$$- \sum_{j}\sum_{t} c\,p_{jt}Y_{jt} - \sum_{t}\left(ch_t \sum_{j}\sum_{k} X_{jkt} \right) - \sum_{t}\left(cr_t \sum_{j}\sum_{k} Z_{jkt} \right) - \sum_{t} F_t\, clabf_t \tag{20}$$
$$- \sum_{t} H\,ire_t clabp_t - \sum_{k}\left(cwaterw_k \sum_{j}\sum_{t} X_{jkt} \right) - \sum_{j}\left(clightingv_j \sum_{k}\sum_{t} X_{jkt} \right)$$

The model is designed to optimize decisions around planting and harvesting to maximize predicted revenues for farmers. This highlights the discrepancy in aspects such as revenue projections from selling to markets (SWM), traders (ST), and byproduct providers, compared to the overall expenditures incurred for rooting, truncating, byproduct processing, penalties for demand shortfall, water costs, laboring, and lighting.

- Constraints: the below constraints are applied annually.
    - Land availability.

Each crop has an expanse of *j* at an age of k that does not exceed the available land (*L*).

$$\sum_{j}\sum_{k} X_{jkt} \leq L \tag{21}$$

- Age-class balance in planting

The planning structure of agricultural models is related to the third category of limitations (22)–(27). Catalá [89] modeled the process of chopping down in a study of planting new apple and pear trees.

Y_{jt} resolves the plantation decisions. A freshly cultivated fruit tree is always considered to be of age 0. Constraint (22) specifies that only crops in age class 1 can be planted in year 1.

$$X_{jkt} = Y_{jt} \quad \forall \begin{cases} 1 \leq j \leq J \\ k = 1 \end{cases} \tag{22}$$

Constraint (23) ensures that the newly planted crops are uncut in the same year.

$$Z_{jkt} = 0 \quad \forall \begin{cases} 1 \leq j \leq J \\ k = 1 \end{cases} \tag{23}$$

Only the first year is subject to constraints (24) and (25), while other age classes ($k = 2 \ldots 10$) are not. Based on this, the planted sector is the number of trees in age class $k - 1$ in year 0 less than the amount that can be felled in the first year once the trees have survived for a year.

$$X_{ijt} = I_{j,k-1} - Z_{jkt} \quad \forall \begin{cases} 1 \leq j \leq J \\ 2 \leq k \leq K \end{cases} \tag{24}$$

$$Z_{jkt} = I_{i,k-1} \quad \forall \begin{cases} 1 \leq j \leq J \\ k = K \end{cases} \tag{25}$$

In the planning horizon, periods $t > 1$ are subject to constraints (26) and (27). For crops with age classes $10 \geq k > 1$, constraint (26) is applicable. Relative to the optionally cleared area, the extent of the plantation each year is determined by the preceding year.

$$X_{jkt} = X_{j,k-1,t-1} - Z_{jkt} \quad \forall \begin{cases} 1 \leq j \leq J \\ 2 \leq k \leq K \end{cases} \tag{26}$$

All crops that reach the age of 9 in the considered year $t - 1$ should be truncated in the following year, according to constraint (27).

$$Z_{jkt} = X_{j,k-1,t-1} \quad \forall \begin{cases} 1 \leq j \leq J \\ k = K \end{cases} \tag{27}$$

- Constraint (28) makes sure that the total harvest in each scenario is less than the product of the planting area (in hectares) by the yield (metric tons/hectare).

$$\sum_i ST_{jits} + \sum_m SWM_{jmts} + SB_{jts} \leq \sum_j y_{jkt} X_{jkst} \quad \forall \begin{cases} 1 \leq j \leq J \\ 1 \leq s \leq S \end{cases} \tag{28}$$

- In each scenario, constraints (29)–(31) set the demand satisfaction in each year.

$$T_{jits} = djit - \epsilon^1_{jits} \quad \forall \begin{cases} 1 \le j \le J \\ 1 \le i \le I \\ 1 \le s \le S \end{cases} \tag{29}$$

$$SWM_{jmts} = ejmt - \epsilon^2_{jmts} \quad \forall \begin{cases} 1 \le j \le J \\ 1 \le m \le M \\ 1 \le s \le S \end{cases} \tag{30}$$

$$SB_{jts} = f_{jt} \quad \forall \begin{cases} 1 \le j \le J \\ 1 \le s \le S \end{cases} \tag{31}$$

- Labor constraints

$$F_t + Hire_t - P_t . \sum_j Y_{jt} - H_t . \sum_j X_{jt} - R_t . \sum_j Z_{jkt} = 0 \qquad \forall t \tag{32}$$

Constraint (32) stands for the labor requirements to plant, cut, and harvest in a given year, in which:

$$F_t = M \qquad \forall t. \tag{33}$$

There may occasionally be a set number of full-time employees. In this case, it is expressed using:

$$Hire_t \le N \qquad \forall t. \tag{34}$$

The requirements of cultivating, harvesting, or truncating could determine the hiring of the part-time workforce. However, as noted in constraint (34), this number is restricted by the limits of the budget.

- Water limitation

$$\sum_j \sum_k X_{jkst} . w_{jkst} \le W_{st} \qquad \forall t, s \tag{35}$$

For each age k of a tree, the water requirement per hectare for a specific crop j varies in season s of year t. The variables must not surpass the water available, since excessive watering of trees leads to decreased output.

- Lighting limitation

$$\sum_{j} X_{js}.v_s \leq V_s \qquad \forall s \qquad (36)$$

7.4.3 Definition of indices, variables, and parameters of the stochastic model

Indices

Symbol	Description	Symbol for Max Value
j	Dragon fruit species	J
k	Age classes	K
i	Traders	I
m	Wholesale markets	M
t	Time periods	T
s	Scenarios	S

Variables

Indexer	Symbol	Description
year, crop, trader, scenario	ST_{jits}	Amount of crop j delivered to trader i in time t for scenario s
year, crop, trader, scenario	ϵ_{jits}^{1}	Amount of crop j under delivered to trader i in time t for scenario s
year, crop, market, scenario	SWM_{jmts}	Amount of crop j delivered to wholesaler m in time t for scenario s
year, crop, market, scenario	ϵ_{jmts}^{2}	Amount of crop j under delivered to wholesaler m in time t for scenario s
year, crop, scenario	SB_{jts}	Amount of crop j harvested for byproducts in time t for scenario s
year, crop, age	X_{jkt}	Area of crop j planted in time t within age class k
year, crop, age	Z_{jkt}	Area cut of crop j of age class k in time t
year, crop	Y_{jt}	Area newly cultivated with crop j in year t
year	F_t	Quantity of permanent employees in t
year	$Hire_t$	Part-time employees recruited during time t

Parameters

Indexer	Symbol	Description
	L	Amount of land available
		Maximum lighting per hectare
		Maximum water per hectare
age	w_k	Water required per hectare for age class k
age	$cwater$	Cost of required water per hectare
crop	v_j	Lighting required per hectare for crop j
crop	$clighting$	Cost of required light per hectare
year	cr_t	Cost of cutting per hectare during time t
year	ch_t	Cost of harvesting per hectare during time t
year	cbp_t	Processing costs per ton during time t

year	$clabf_t$	Periodic cost of fixed staff
year	$clabp_t$	Cost of labor for part-time employees each period
year	R_t	Number of employees required to cut one hectare
year	H_t	Number of employees required to harvest one hectare
year	P_t	Number of employees required to plant one hectare
crop, age	I_{jk}	Initial area of crop j of age class k
year, crop	f_{jt}	Demand for byproducts of crop j in period t
year, crop	u_{jt}	Minimum planting area per crop j in period t
year, crop	cp_{jt}	Cost per hectare of planting for crop j in period t
year, scenario	ρ_s	Estimated price probability of scenario s
year, crop, age	y_{jkt}	Production in tons per hectare of crop j in age class k during the given period t
year, crop, market	e_{jmt}	The wholesale market's demand for crop j during that time t
year, crop, market	$cPNT2_{jmt}$	Penalty for wholesaler m not satisfying demand for each ton of crop j during period t
year, crop, scenario	r_{jts}	Price per ton of byproducts in period t in scenario s
year, crop, trader	d_{jit}	The trader's demand for crop j during that time t
year, crop, trader	$cPNT1_{jit}$	Penalty for trader i not satisfying demand for each ton of crop j during period t
year, crop, market, scenario	p_{jits}	For trader i in period t in scenario s, the price per ton of crop j
year, crop, trader, scenario	q_{jmts}	For wholesaler m in period t in scenario s, the price per ton of crop j

7.4.4 Scenario Tree Generation

The method of generating scenario trees is a highly effective tool in the field of stochastic programming, and it has been extensively employed in the resolution of various practical issues. The method in question possesses characteristics that render it adaptable, interpretable, accurate, and scalable.

The utilization of the scenario tree generation technique is prevalent in various domains such as finance, operations research, and strategic planning. This tool facilitates the ability of decision-makers to create models that account for uncertainties, generate estimations of potential outcomes, and enhance the process of decision-making across a range of conditions. The process entails the creation of a hierarchical arrangement characterized by interconnected nodes, wherein each node corresponds to a distinct decision, event, or consequence.

The methodology was initially presented by Howard Raiffa in his publication titled "Decision Analysis: Introductory Lectures on Choices under Uncertainty" [167]. The concept has been subsequently extended and improved upon by several academics, such as Dantzig and Infanger [168] who employed the technique to address complex linear programming issues on a significant scale.

In a study, Calfa et al. [169] introduced a novel approach to construct multi-stage scenario trees that maintain consistency with both historical and projected data. The methodology employed in this study follows a two-step process. Firstly, statistical property matching is utilized to develop a collection of scenarios that exhibit statistical features similar to those observed in historical data. Secondly, a Distribution Matching Problem is addressed to verify that the generated scenarios also align with the forecasted distribution of the data. The method suggested in this study presents a novel and effective approach for generating scenario trees in the context of multi-stage stochastic programming issues. The methodology is based on robust statistical principles and demonstrates computational efficiency. Consequently, it will be selected for implementation in the context of optimizing the production and distribution processes pertaining to fresh fruit in Vietnam.

In this chapter, based on the algorithm proposed by Calfa et al. [169], a Distribution Matching Problem (DMP) model is developed using the Moment Matching Problem combined with the Empirical Cumulative Distribution Function (ECDF) and is designed to be applied to build scenario trees based on the fresh fruit price dataset collected in Vietnam. The procedure of the algorithm and an illustrated scenario tree are depicted in Figures 7.7 and 7.8 below.

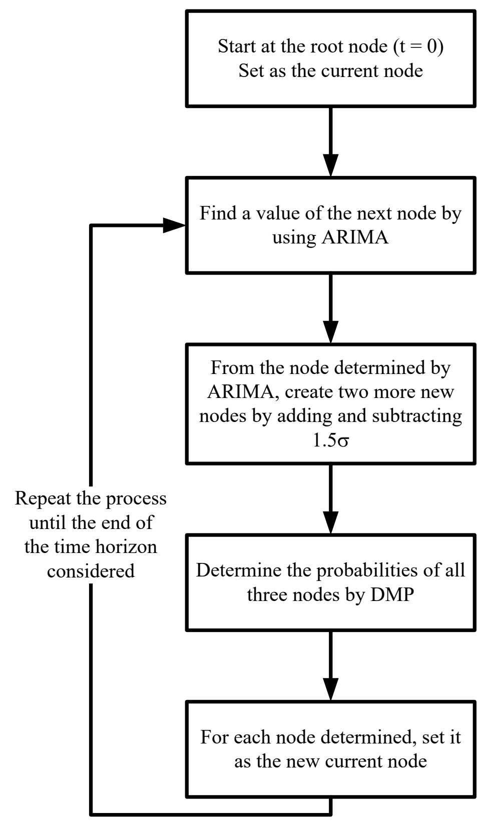

Figure 7.7 The procedure of building a scenario tree for the selling price of the dragon fruits.

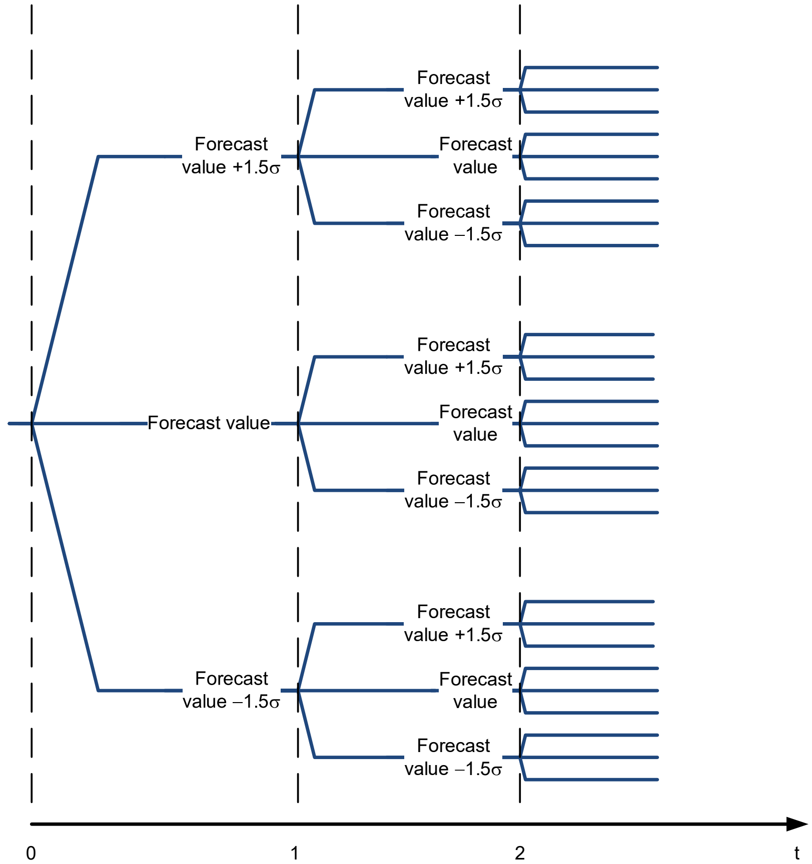

Figure 7.8 Illustrated scenario tree for the selling price of the dragon fruits.

The objective of using the Distribution Matching Problem is to determine the ideal values of random variables and probabilities associated with scenario trees, with the aim of minimizing discrepancies between statistical features derived from the tree and those computed directly from the available data [169]. The mathematical model of the DMP is described in Appendix B4.

7.4.5 Sample Average Approximation

In stochastic optimization issues where the objective function cannot be determined precisely but can be estimated through simulation, sample average approximation (SAA) is a common approach. To solve the approximate problem using deterministic optimization techniques, the original problem is replaced by an approximation based on a finite sample of random scenarios. With independent training samples, SAA has strong asymptotic performance guarantees, but these assurances might not be universally true with dependent samples [170, 171].

The SAA method in our study uses Monte Carlo simulation to address optimization problems with stochastic elements. In this approach, the predicted objective function is estimated by calculating the sample mean from a random sample [172]. To deal with the fresh fruit production and distribution problem, the SAA operates by producing a collection of scenarios that accurately depict the outcomes of uncertain price fluctuations.

The procedure for applying SAA is described in Figure 7.9, as follows.

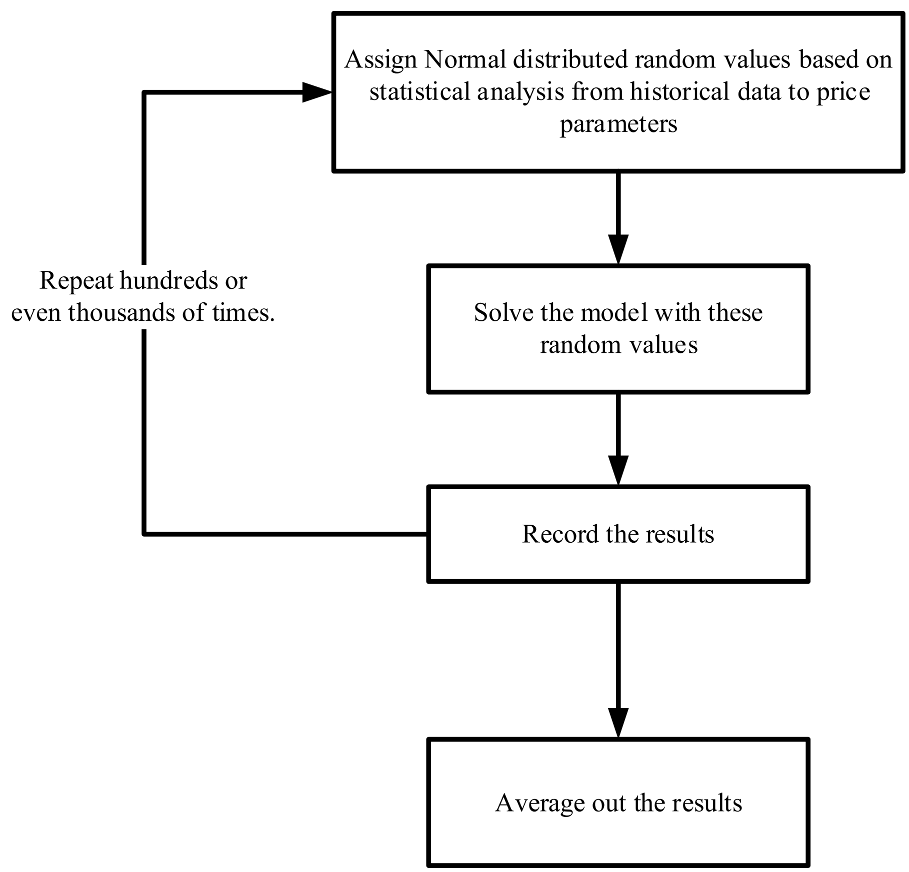

Figure 7.9 The procedure for applying SAA for the production and distribution of fresh fruit stochastic model.

7.4.6 Chance-Constrained Programming

Chance-constrained programming (CCP) is a robust and adaptable optimization technique utilized to formulate and address stochastic programming issues in the presence of uncertainty [173]. The optimization framework is enhanced by the inclusion of probabilistic constraints, which guarantee that the solution adheres to a specified probability threshold for constraint satisfaction [174].

The idea of CCP was initially mentioned by Markowitz [174] to address portfolio optimization issues. In their seminal paper on optimizing oil distribution systems, Charnes and Cooper [175] formally established CCP, marking a significant milestone in the field of stochastic optimization. Subsequently, this concept has been expanded to encompass a wide range of applications, such as energy management, supply chain planning, financial risk management, and project scheduling.

The issue of fulfilling demand was resolved with the implementation of a penalty system for any instances of insufficient supply. Our objective is to ensure that the requirements of every product from each dealer are met with a probability that exceeds a specified level of reliability (*Rel*). The concept is denoted as a chance constraint.

Let Rel be a desirable dependability value, where $0 \leq Rel \leq 1$. The presence of shortage factor ε is no longer necessary; yet it remains crucial to ascertain whether the demand will be satisfied or not. In the context of the chance-constrained deterministic formulation, it is necessary to introduce binary variables to accurately describe the chance constraint. In this section, therefore, we redefine ϵ as a binary variable.

- ϵ^1_{jits} is 1 if a shortage of crop *j* happens at trader *i* in period *t* for scenario *s,* 0 otherwise.
- ϵ^2_{jmts} is 1 if a shortage of crop *j* happens at wholesaler *m* in period *t* for scenario *s*, 0 otherwise.

Since the deficiency penalty no longer needs to be minimized, the new objective function (37) developed and modified from the objective function (1) is introduced as follows:

$$\max \quad \sum_s \rho_s \Big(\sum_j \sum_i \sum_t p_{jits}\, ST_{jits} + \sum_j \sum_m \sum_t q_{jmts}\, SWM_{jmts} + \sum_j \sum_t r_{jts}\, SB_{jts} - \sum_t \Big(cbp_t \sum_j S\, B_{jts} \Big) \Big)$$
$$- \sum_j \sum_t c\, p_{jt} Y_{jt} - \sum_t \Big(ch_t \sum_j \sum_k X_{jkt} \Big) - \sum_t \Big(cr_t \sum_j \sum_k Z_{jkt} \Big) - \sum_t F_t\, clabf_t$$
$$- \sum_t H\, ire_t clabp_t - \sum_k \Big(cwaterw_k \sum_j \sum_t X_{jkt} \Big) - \sum_j \Big(clightingv_j \sum_k \sum_t X_{jkt} \Big) \tag{37}$$

Furthermore, to ensure the attainment of the necessary level of reliability, the following constraints are incorporated:

$$\textstyle\sum_s \rho_s . \epsilon^1_{jits} \leq 1 - Rel^1 \quad \forall \begin{cases} 1 \leq j \leq J \\ 1 \leq i \leq I\ ; \\ 1 \leq s \leq S \end{cases} \tag{38}$$

$$\textstyle\sum_s \rho_s . \epsilon^2_{jmts} \leq 1 - Rel^2 \quad \forall \begin{cases} 1 \leq j \leq J \\ 1 \leq m \leq M. \\ 1 \leq s \leq S \end{cases} \tag{39}$$

However, a limitation of the above constraints is the absence of a guarantee that variables ϵ will assume a value of 1 in the event of a shortage. Constraints (29) and (30) will be rewritten to satisfy the requirement that, when there is a shortage, the variable ϵ will have the value 1:

$$d_{jit} - ST_{jits} = 10{,}000 \epsilon^1_{jits} \quad \forall \begin{cases} 1 \leq j \leq J \\ 1 \leq i \leq I\ , \\ 1 \leq s \leq S \end{cases} \tag{40}$$

$$e_{jmt} - SWM_{jmts} = 10{,}000 \epsilon^2_{jmts} \quad \forall \begin{cases} 1 \leq j \leq J \\ 1 \leq m \leq M. \\ 1 \leq s \leq S \end{cases} \tag{41}$$

Constraints (21)–(28) and (31) of the stochastic model remain unchanged.

Finally, the rewritten model with probabilistic constraints is solved by using the sample average approximation method with 200 iterations.

The case study aims to optimize the production and distribution of dragon fruit from growers to traders, wholesalers, and byproducts, with the objective of maximizing profits. To achieve this, three approaches are employed: scenario tree generation, sample average approximation, and chance-constrained programming. These methodologies are utilized to identify potential solutions for the case study. The results obtained from the used methodologies are presented and analyzed to provide a comprehensive conversation. This discussion aims to offer help and guidance to growers or managers involved in the production of dragon fruit, aiding them in making informed decisions.

The stochastic model was coded and solved in Python language (Appendix C2, C3, C4 and C5) on a computer configured with a 12th Gen Intel (R) Core (TM) i7 Processor and RAM of 32.0 GB.

7.5 RESULTS AND DISCUSSION

The results herein obtained are, in fact, the extension study of the scenarios deterministically solved and presented in Nguyen et al. [41]. These findings can assist farmers in making decisions regarding the allocation of land for several varieties of dragon fruit throughout a 20-hectare area over an 8-year period with fluctuating selling prices. As mentioned above, selling prices of dragon fruits completely plunged during the outbreak of the COVID-19 pandemic; the question "how dragon fruit growers would act if a similar thing were to happen, and how they could maximize their profit as well as minimize their risks in the future", requires further investigation.

According to Nguyen et al. [41], three kinds of dragon fruit are planted for local demands and exporting including red-skin, white-flesh (Crop 1); red-skin, red-flesh (Crop 2); and yellow-skin, white-flesh (Crop 3). Since China is the largest dragon fruit importer market, the prices are dominated mostly by Chinese traders. The selling price agreed between the growers and the traders is mainly based on trust (verbal contract); if the trader terminates the contract because of finding a better source of dragon fruits, the farmer will suffer.

Before the COVID-19 pandemic in January 2020, the price of Crop 1 remained stationary during episode 1 (peak season) and slightly rose in episode 2 (off-peak season). Crop 2 was priced double as high due to its cultivation for sale to the Chinese markets, where there is an extremely high demand. The price of Crop 3 cultivated in Vietnam was below expectations. Vietnamese farmers then aimed to sell their locally grown fruits at a price equivalent to the selling price of Crop 3 imported from Malaysia, which is 20 times greater than white-flesh dragon fruits. Its greatest price is equivalent to the price of red-flesh ones.

During the pandemic period, the prices of all three kinds of dragon fruit dramatically fell when the lockdowns were declared due to the outbreaks. Many farms did not have traders, so ripe dragon fruits were used as food for cattle or on a big sale— USD 0.1–0.2 per kilogram. At that selling price, the growers do not have enough revenue to compensate for production costs for the next season.

The current study presumes that Crop 1 is the established supply chain for both local and international trade with a recognized measurable demand. Crop 2 is exclusively cultivated for importation into Chinese markets, whereas Crop 3 is grown to assess its market potential. The selling prices of all crops in the model were assumed to randomly vary following a normal distribution curve over the years.

Establishing a specific land ratio for different types of dragon fruit is envisaged to help dragon fruit growers protect their profits from market price fluctuations caused by such COVID-19 pandemic events or severe unpredicted weather conditions resulting from climate change in the next several years.

7.5.1 Scenario Tree Generation

A look at historical data shows the variation in selling prices of red-skinned, white-fleshed dragon fruit and red-fleshed, red-skin dragon fruit, with monthly average data from July 2019 to December 2022 (Figure 7.10) and average annual data from 2013 to 2022 (Figure 7.11).

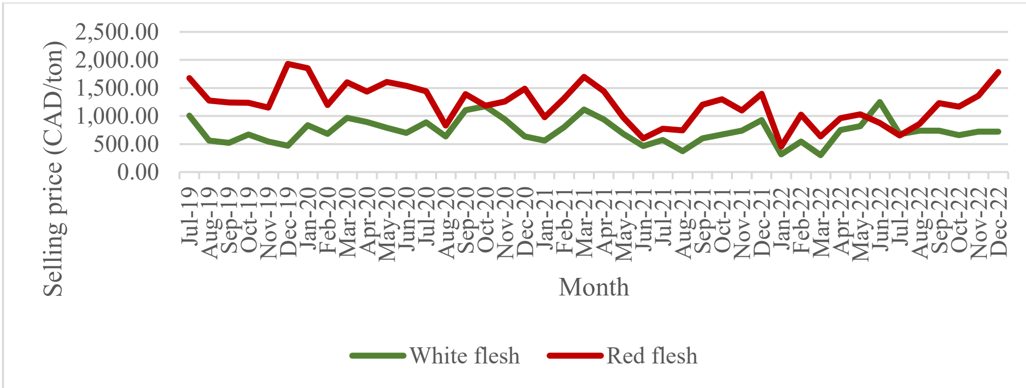

Figure 7.10 Historic data of monthly selling prices of white-flesh, red-skin and red-flesh, red-skin DF.

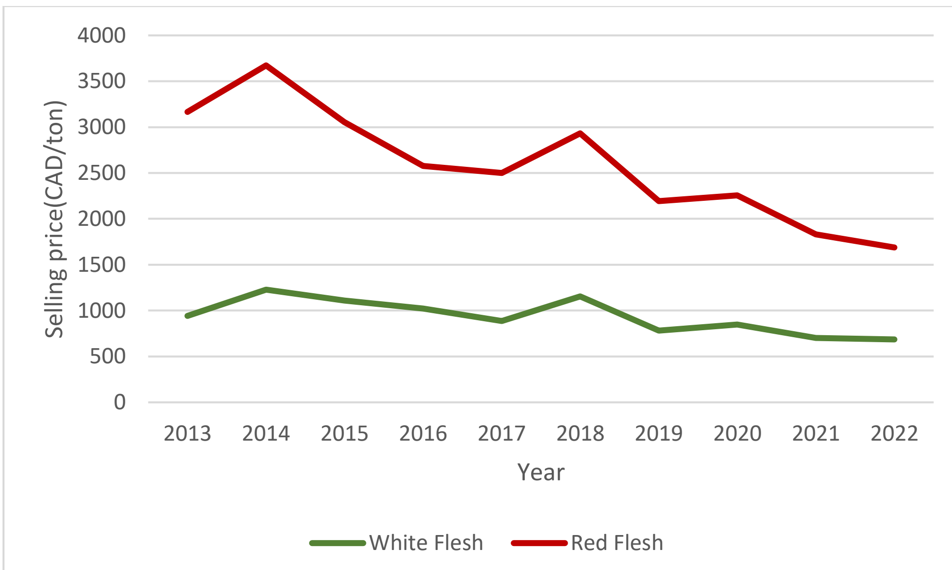

Figure 7.11 Historic data of yearly selling prices of white-flesh, red-skin and red-flesh, red-skin DF.

An uncomplicated method for constructing the framework of a multi-period scenario tree is to have three potential outcomes at each node. These outcomes consist of the predicted value as well as two values obtained by adding and subtracting 1.5 standard deviation from the predicted value for each period. The predicted value is derived from the ARIMA model, with a 95% level of confidence and the parameters p, d, and q set to 1, 0, and 0, respectively. Since the selling price data are applied based on the ARIMA model, it is expected that the forecasted value will conform to a normal distribution. The standard deviation (σ) value is derived from statistical analysis of historical data on dragon fruit selling prices in Binh Thuan, Vietnam. There are eight time periods corresponding to the annual production planning problem over an eight-year period. Hence, the number of possible scenarios for planning dragon fruit production and distribution over an 8-year period is 6561 (3 raised to the power of 8). The variability in the selling price of the three different varieties of dragon fruit cultivated and consumed in Vietnam is considered: white-flesh, red-skinned dragon fruit; red-flesh, red-skinned dragon fruit; and white-flesh, yellow-skinned dragon fruit. The scenario tree for each selling price is assessed separately and subsequently merged into a unified tree, as shown in Figure 7.12.

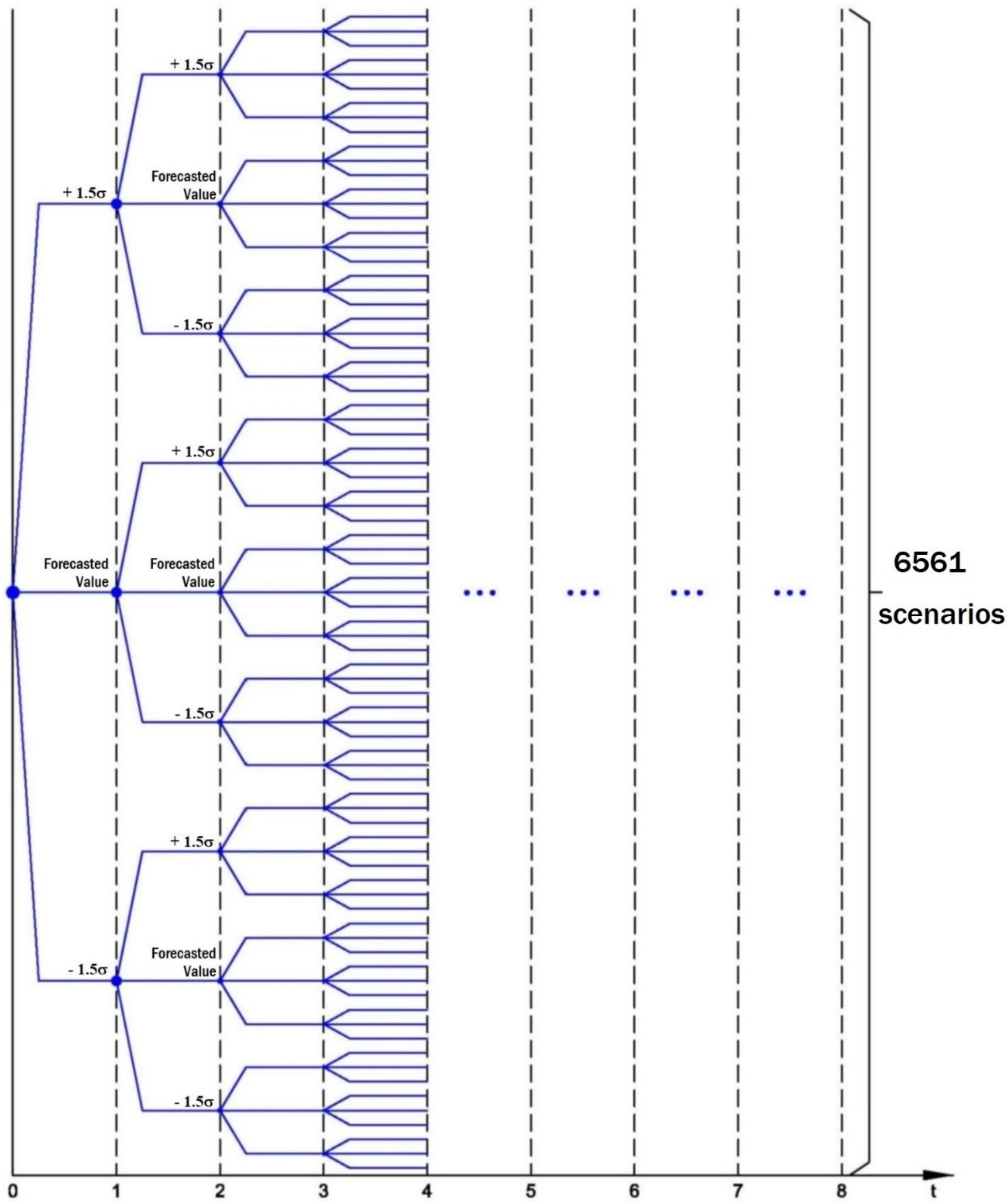

Figure 7.12 Scenario tree for selling prices of dragon fruits in an 8-year period.

In order to proceed, it is necessary to ascertain the probability associated with each potential outcome of the scenario tree by utilizing the Distribution Matching Problem (DMP) model that was presented in the preceding section. The statistical parameters utilized in the DMP model, including the mean value, standard deviation, variance, and covariance, are derived from the ARIMA forecasting model introduced in the preceding stage. The likelihood value was calculated promptly upon acquiring the forecast value, upper value, and lower value. Hence, a total of 3280 iterations of the DMP model were conducted to determine the probability of all possible outcomes in the scenario tree over the course of the 8-year study period. An assumption was made that the chance of the selling price being the same for each outcome is equal for all three types of dragon fruit.

The stochastic programming model was implemented and solved using the Python programming language, resulting in a value of $12,603,389 after a computation time of around 12,916.88 seconds.

7.5.2 Sample Average Approximation

The sample average approximation (SAA) is a method for solving stochastic optimization issues by utilizing Monte Carlo simulation. Given the extensive research period and the large number of scenarios involved in determining the optimal model for the production and distribution of fresh dragon fruit in Vietnam, the Monte Carlo sampling method is a prudent choice. This method is both simple and reliable, as the results obtained from the sample average approximation (SAA) are addressed using deterministic optimization techniques.

Normal distribution was found to be followed by the selling price of dragon fruit, as can be seen from the historical data that was collected. The variable that represents the selling price of three different kinds of dragon fruit should be created and given random values in accordance with the Normal distribution. This objective function of the model should be computed using these random values, and the results should be recorded. The random values of the variables should be regenerated and reassigned. Perform another calculation of the objective function. Repeat the processes that were just described many times and then determine the average. The iterations of 20 are the ones that are suggested in this study. The results of the model are presented in Table 7.2 and Figure 7.13.

Table 7.2 Results of SAA model

Number of scenarios		100	200	500	1000	2000	4000	6561
Elapsed Time		3.57	11.9	55.8	301.13	938.03	4266.2	12837.36
Constraints		29096	57896	144296	288296	576296	1152296	1889864
Variables		50906	101306	252506	504506	1008506	2016506	3307250
Profit								
	Min	11,759,872	11,745,591	11,944,122	11,876,687	11,793,708	11,873,843	11,824,094
	Max	12,399,518	12,420,237	12,311,794	12,240,204	12,171,073	12,204,516	12,138,131
	Mean	12,058,235	12,081,421	12,116,350	12,027,145	11,998,156	12,015,784	11,980,572

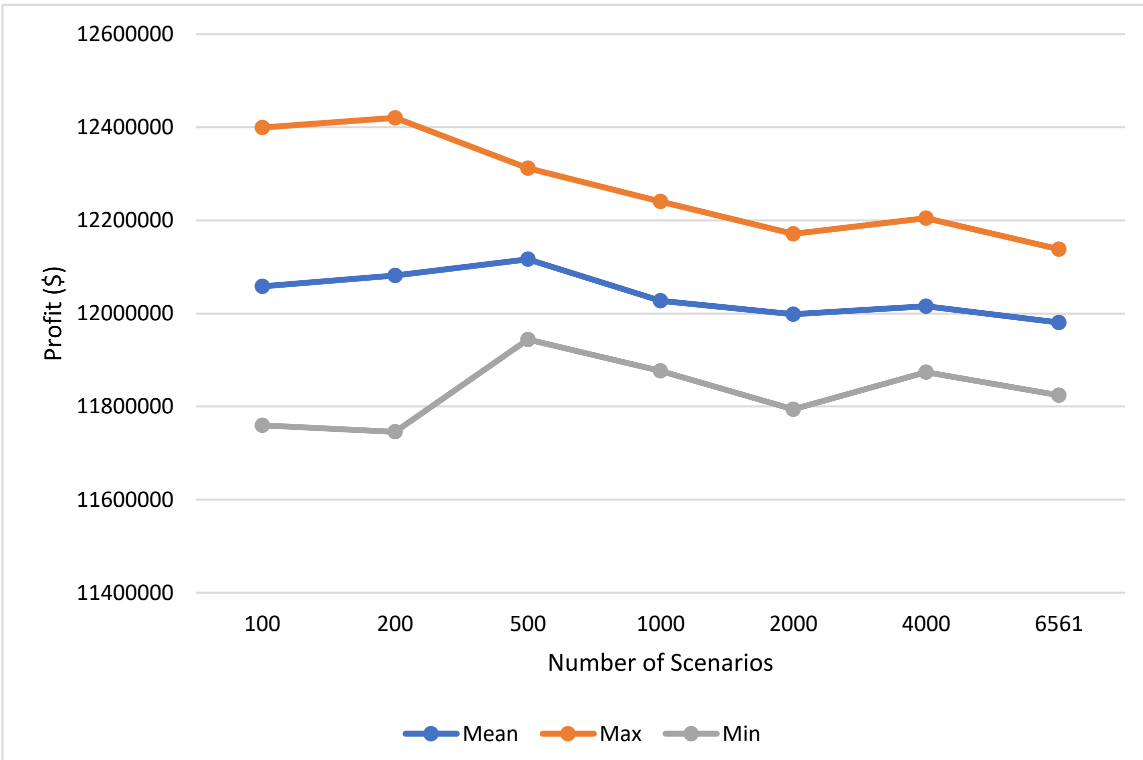

Figure 7.13 Results of Sample Average Approximation with different numbers of scenarios

7.5.3 Chance-Constrained Programming

The objective function of the TSSP model that is introduced in the previous section has been rewritten in order to optimize the production and distribution of dragon fruit in Vietnam. Additionally, probabilistic constraints are added to specify the minimum probability with which the solution should satisfy the original constraints. Since prices vary, we looked at Chance-Constraint optimization to evaluate the objective for guaranteed price threshold percentiles of 25, 50, 75, and 99. As can be seen from Figure 7.14, the value of 50 percentile price guarantee yields a solution comparable to the deterministic model.

Figure 7.14 Objective function results of chance-constrained programming with different selling price percentiles.

The most optimal outcomes derived from those mentioned methodologies or approaches are briefly presented in Table 7.3.

Table 7.3 Summary of results

	Deterministic Model (Nguyen et al., 2020) [13]	**Deterministic Model (Python)**	**Scenario Tree Generation**	**Sample Average Approximation with 6561 scenarios (20 Iterations)**	**Chance-Constrained Programming with 50% price percentile**
Objective function	$11,342,442	$11,376,580	$ 12,603,389	$ 11,980,572	$11,652,642
Number of variables	2510	1024	3,307,250	3,307,250	3,307,264
Number of constraints	2387	608	1,889,864	1,889,864	3,464,528
Running time	0.1 second	0.06 second	12,916.88 seconds	12,837.36 seconds	4,478 seconds

Based on the results shown in Table 7.3, it is evident that the stochastic programming model outperforms the linear programming deterministic model during the period of 8 years. Additionally, some extra derived metrics are calculated and shown in Table 7.4.

Table 7.4 Some extra derived metrics to compare among models

	Deterministic (Python)	Scenario Tree Generation	Sample Average Approximation	Chance-Constrained

				Programming with 50% of price percentile
Total profit	$ 11,376,580	$ 12,603,389	$ 11,980,572	$ 11,652,642
Total cost	$ 4,239,680	$ 4,079,108	$ 4,640,005	$ 2,757,558
Total yield	9,600 (tons)	9600 (tons)	9,600 (tons)	9600 (tons)
Average yield/ha/year	60 (tons)	60 (tons)	60 (tons)	60 (tons)
Average cost/ha/year	$ 26,498	$ 25,494	$ 29,000	$ 17,234
Average cost/kg/year	$ 0.44	$ 0.425	$ 0.48	$ 0.29
Average profit/ha/year	$ 71,103.63	$ 78,771	$ 73,571	$ 72,829
Average profit/kg/year	$ 1.19	$ 1.31	$ 1.23	$ 1.21

7.6 Conclusions

This work introduces a stochastic model for planning and organizing the production and distribution of dragon fruit in Vietnam, taking into account the unpredictable fluctuations in selling prices. Two methods, scenario tree generation and sample approximation average, are used to address the uncertainty of the issue by estimating the expected value of the objective function. Furthermore, a very effective and adaptable optimization approach called chance-constrained programming is suggested to consider the uncertainty in the dragon fruit trading price and its impact on meeting consumer demand.

Our stochastic approach in this chapter encompasses the benefits of the linear programming model proposed by Nguyen et al. [41], which involves deciding whether to cultivate dragon fruit when its selling price is high or rising and discarding dragon fruits with lower prices or that are old. This stochastic model can additionally address the limitation of the deterministic model by effectively handling the unpredictability and ambiguity associated with dragon fruit selling prices. This helps dragon fruit producers and managers gain a more comprehensive understanding when making choices on the selection of dragon fruit types and cultivation areas in medium- and long-term plans.

Nevertheless, this chapter does possess some constraints. Our research does not consider random elements such as demand and yield due to the challenges associated with data collection. Furthermore, the forecasting technique used for the stochastic approach is simplistic and lacks a high level of reliability.

Investigating the production and distribution of dragon fruit, as well as other fresh fruits, remains arduous and challenging. In the future, reliable sample and forecasting techniques will be used to enhance the accuracy of planning. In addition, other stochastic methodologies are being investigated and implemented to provide decision-makers with more valuable information or a wider range of possibilities to consider when formulating long-term strategies.

CHAPTER 8 CONCLUSIONS AND FUTURE PERSPECTIVES

8.1 SUMMARY AND CONCLUSIONS

- This thesis proposed and developed a decision support tool for long-term planning of activities required in the production and distribution of fresh agricultural products. The biggest goal of the research carried out throughout is to increase profits for producers, the first and most important link in the supply chain of fresh agricultural products. Besides, effective planning and management of activities related to the production and distribution of fresh agricultural products increases the sustainability of the entire supply chain. Through mathematical modelling, the study examines and addresses key issues in fresh agricultural planning, including:
  - Planning within limited resources.
  - Production depends on changing weather.
  - High market fluctuations.
  - Perishability of agricultural products.
  - Risk-based planning.
- The thesis objectives outlined in this thesis were successfully accomplished in the following manner:
  - A complete literature study was undertaken and published in the journal AgriEngineering in July 2021 to showcase the indispensability and significance of mathematical models in fresh fruit supply chain management. By examining the positive and negative aspects, as well as the limitations and benefits of mathematical models built by other researchers, I have devised and constructed mathematical models and quantitative methods that are specifically tailored to address the uncertainty and risks associated with the fresh fruit supply chain in Vietnam.
  - A Mixed Integer Programming model was developed to maximize the cultivation and distribution of dragon fruit in Vietnam by considering hypothetical situations of price changes. Nevertheless, the deterministic model fails to comprehensively account for unpredictable variables such as market price fluctuations, customer

demand fluctuations, and the impact of climate change. The findings of this study were published in the scientific journal AgriEngineering in December 2019.

  - The stochastic optimization model is designed to tackle the complex impacts of uncertain factors that cannot be resolved in the deterministic model for fruit production and distribution in Vietnam, using the specific case of dragon fruit. Three robust and efficient methods, namely scenario tree generation, sample average approximation, and chance constrained programming, have been selected to address the stochastic problem. The paper was reviewed and published by Sustainability magazine in March 2024.

- The benefits of the thesis via the development of mathematical models for the application of planning tool are:
  - The thesis created a comprehensive hierarchical planning system for the tactical and operational planning of perishable agricultural goods. The efficacy of employing operations research models for the planning of fresh fruit, such as dragon fruit, was demonstrated by this planning system. The suggested decision support system offers the advantage of effectively managing risk, coordinating short-term and long-term decisions, and balancing the trade-offs between costs and the quality of fresh fruit.
  - The concept includes not just deterministic approaches to planning. The research also incorporates a stochastic model that addresses the inherent uncertainty in prices. To solve the stochastic model, I suggested employing robust and efficient approaches such as scenario tree generation, sample average approximation, and chance constrained programming. These techniques can help in obtaining faster answers.
  - A significant contribution of the thesis is the implementation of the stochastic model in the tactical planning of fresh agricultural goods. This model can be utilized in the future to assess the advantages of various options for mitigating risks. This methodology can assist producers in making comprehensive decisions on risk and anticipated revenues.
  - Another key contribution of this thesis is the development of a methodology for researching, designing, and implementing models using real data and case studies.

The models created offer substantial benefits to both growers and decision makers in the fresh fruit supply chain.

8.2 FUTURE PERSPECTIVES

There are still a lot of limitations to the research that was done throughout the thesis. As a result of the difficulties connected with data collecting, our research does not consider random factors like demand and yield. In addition to this, the method of forecasting that is used for the stochastic approach is very simple and does not possess a high degree of reliability.

Continued investigation into the production and marketing of dragon fruit, in addition to other perennial fresh fruits that develop quickly, continues to be a difficult and tough endeavour. It is anticipated that in the future, methodologies for trustworthy sampling and forecasting would be used to improve the accuracy of planning. The ability to estimate market prices and consumer demand is an essential tool for decision makers who deal with fresh product such as fruits. This is because these techniques assist to reduce risks, improve financial performance, increase customer satisfaction, and gain a competitive edge in a market that is both dynamic and unpredictable.

Furthermore, in addition to the price changes that are taken into consideration in this thesis, the demand for fresh fruit and the output of fresh fruit are both vulnerable to a variety of uncertainties that may have a substantial influence on both the producers and the consumers. There are several unknown elements that influence client requirements, including consumer tastes, economic situations, competition, worries over food safety, and seasonality. As variable determinants of yield, it is important to take into consideration the unpredictability of elements such as the weather, agricultural pests and diseases, pollination, and crop management strategies.

Last but not least, more stochastic techniques or robust approaches are now being researched and put into practice in order to provide decision-makers with more important information or a larger variety of alternatives to take into consideration when creating long-term policies.

REFERENCES

1. Anh-Tuyet; Hoang-Hung; Dang-Giang; Huu-Nghia;Nguyen-Su. Đổi mới tư duy sản xuất và tiêu thụ trái cây để khai phá tiềm năng. *Nhandan.vn*. Hanoi, Vietnam. 2023. (accessed on April 05, 2024).

2. Vu, V. T. *BÁO CÁO TRIỂN VỌNG NGÀNH HÀNG TRÁI CÂY TẠI VIỆT NAM 2024*. *Kirin Capital*. Hanoi, Vietnam. 2024. https://kirincapital.vn/vi/bao-cao-trien-vong-nganh-hang-trai-cay-viet-nam-2024/. (accessed on

3. MARD. Phê duyệt Đề án Phát triển cây ăn quả chủ lực đến năm 2025 và 2030 (4085/QĐ - BNN - TT). Hanoi, Vietnam. 2022. Available online: https://thuvienphapluat.vn/van-ban/Linh-vuc-khac/Quyet-dinh-4085-QD-BNN-TT-2022-phe-duyet-De-an-Phat-trien-cay-an-qua-chu-luc-den-2025-535508.aspx.

4. MondorIntelligence. *Vietnam Fruits and Vegetables Market Size & Share Analysis - Growth Trends & Forecasts (2024 - 2029)*. India. 2023. https://www.mordorintelligence.com/industry-reports/vietnam-fruits-and-vegetables-market. (accessed on

5. Nguyen, H. T.;Degenhardt, P. The Many Climate Challenges Facing the Mekong Delta. *Rosa-Luxemburg-Stiftung*. Berlin, Germany. 2021. https://www.rosalux.de/en/news/id/44262/the-many-climate-challenges-facing-the-mekong-delta (accessed on April 05, 2024).

6. IMHEN; Committee, C. M. P.;Committee, K. G. P. *Climate Change Impact and Adaptation Study in The Mekong Delta – Part A Final Report: Climate Change Vulnerability and Risk Assessment Study for Ca Mau and Kien Giang Provinces, Vietnam*. *Institute of Meteorology, Hydrology and Environment (IMHEN)*. Hanoi, Vietnam. 2011. https://www.adb.org/projects/documents/climate-change-impact-and-adaptation-study-mekong-delta-final-report. (accessed on

7. Ngo-Duc, T.; Nguyen, M.-H.; Pannier, E.; Woillez, M.-N.; Drogoul, A.; Huynh, T. P. L.; Le, T. T.; Nguyen, T. T. H.; Nguyen, T. T.; Nguyen, T. A.; Thomas, F.; Truong, C. Q.; Vo, Q. T.;Vu, C. T. *Climate change in Viet Nam; Impacts and adaptation. A COP26 assessment report of the GEMMES Viet Nam project*. *Agence Française de Développement*. Paris, Fance. 2021. (accessed on

8. Viet-Hoang. Vượt qua mùa hạn mặn nhất lịch sử, những bài học cho hiện tại, tương lai. *National Steering Committee for Disaster Prevention and Control*. Hanoi, Vietnam. 2020. (accessed on April 05, 2024).

9. Huyen-Tran. Miền Tây khát: 'Nước giờ còn quý hơn vàng'. *BBC News Tiếng Việt*. London, England. 2024. https://www.bbc.com/vietnamese/articles/c9rzpyz9110o (accessed on April 10, 2024).

10. Le, H. Chi phí logistics 'ăn mòn' lợi nhuận của nông sản xuất khẩu. *Saigon Times*. Ho Chi Minh City, Vietnam. 2023. https://thesaigontimes.vn/chi-phi-logistics-an-mon-loi-nhuan-cua-nong-san-xuat-khau/ (accessed on April 05, 2024).

11. Viet-Hang. Gắn kết hiệu quả với hệ thống logistics, nâng cao chuỗi giá trị nông sản xuất khẩu. *Tạp chí Công Thương*. Hanoi, Vietnam. 2023. https://tapchicongthuong.vn/bai-viet/gan-ket-hieu-qua-voi-he-thong-logistics-nang-cao-chuoi-gia-tri-nong-san-xuat-khau-106524.htm (accessed on April 05, 2024).

12. Nguyen, V. K. Cây Thanh Long. *Institute of Biotechnology and Environment - Nong Lam University Ho Chi Minh City*. Ho Chi Minh City, Vietnam. 2008. https://rttc.hcmuaf.edu.vn/rttc-8137-1/vn/-cay-thanh-long.html (accessed on October 28, 2015).

13. Phan, T. T. H. The Dragon Fruit Export Challenge and Experiences in Vietnam. *Food and Fertilizer Technology Center for the Asian and Pacific Region*. Taipei, Taiwan. 2019 https://ap.fftc.org.tw/article/1598 (accessed on April 05, 2024).

14. Hoang-Hung. Tăng sức cạnh tranh cho quả thanh long xuất khẩu. *nhandan.vn*. Hanoi, Vietnam. 2023. https://nhandan.vn/tang-suc-canh-tranh-cho-qua-thanh-long-xuat-khau-post777165.html (accessed on April 05, 2024).

15. Minh-Tri. Long Định 1 - "Hoàng hậu" thanh long. *Báo Nông Nghiệp Việt Nam*. Hanoi, Vietnam. 2009. https://nongnghiep.vn/long-dinh-1---hoang-hau-thanh-long-d35704.html (accessed on October 28, 2015).

16. Ky-Phuong;Huu-Danh. Ồ ạt chặt thanh long ruột trắng trồng ruột đỏ: 'Đánh bạc' với thị trường. *Báo điện tử Dân Việt*. Hanoi, Vietnam. 2016. https://danviet.vn/o-at-chat-thanh-long-ruot-trang-trong-ruot-do-danh-bac-voi-thi-truong-7777656822.htm (accessed on March 12, 2016).

17. Nam-Binh. Nearly 1.2 million VND per kilogram of yellow-skin dragon fruit. *Saigon Times*. Ho Chi Minh City, Vietnam. 2020. https://www.sgtiepthi.vn/gan-12-trieu-dong-mot-kilogram-thanh-long-vo-vang/ (accessed on April 05, 2024).

18. Thanhlongviet. Hướng đi mới xuất khẩu thanh long thị trường EU. *Thanh long Viet*. Tien Giang, Vietnam. 2024. https://thanhlongvietnam.vn/huong-di-moi-xuat-khau-thanh-long-thi-truong-eu/ (accessed on).

19. Nguyen, B. T.; Nguyen, T. H. M.;Nguyen, T. M. T. *Bộ Cẩm nang hướng dẫn xuất khẩu vào thị trường Trung Quốc cho sản phẩm trái cây tươi*. *Vietnam Trade Promotion Agency*. Hanoi, Vietnam. 2018. https://trungtamwto.vn/an-pham/13645-cam-nang-xuat-khau-thuc-pham-sang-trung-quoc. (accessed on

20. Caixeta-Filho, J. V. Orange harvesting scheduling management: a case study. *Journal of the Operational Research Society*. **2006**, 57(6), 637-642. DOI: https://doi.org/10.1057/palgrave.jors.2602041.

21. Hamer, P. J. A decision support system for the provision of planting plans for Brussels sprouts. *Computers and electronics in agriculture*. **1994**, 11(2-3), 97-115. DOI: https://doi.org/10.1016/0168-1699(94)90001-9.

22. Widodo, K. H.; Nagasawa, H.; Morizawa, K.;Ota, M. A periodical flowering–harvesting model for delivering agricultural fresh products. *European Journal of Operational Research*. **2006**, 170(1), 24-43. DOI: https://doi.org/10.1016/j.ejor.2004.05.024.

23. Maia, L. O. A.; Lago, R. A.;Qassim, R. Y. Selection of postharvest technology routes by mixed-integer linear programming. *International Journal of Production Economics*. **1997**, 49(2), 85-90. DOI: https://doi.org/10.1016/S0925-5273(96)00108-9.

24. Ferrer, J.-C.; Mac Cawley, A.; Maturana, S.; Toloza, S.;Vera, J. An optimization approach for scheduling wine grape harvest operations. *International Journal of Production Economics*. **2007**, 112(2), 985-999. DOI: https://doi.org/10.1016/j.ijpe.2007.05.020.

25. Masini, G. L.; Blanco, A. M.; Petracci, N.;Bandoni, J. A. Supply chain tactical optimization in the fruit industry. *Process Systems Engineering: Supply Chain Optimization*. **2007**, 4, 121-172.

26. Ahumada, O.;Villalobos, J. R. Operational model for planning the harvest and distribution of perishable agricultural products. *International Journal of Production Economics*. **2011a**, 133(2), 677-687. DOI: https://doi.org/10.1016/j.ijpe.2011.05.015.

27. Ahumada, O.;Villalobos, J. R. A tactical model for planning the production and distribution of fresh produce. *Annals of Operations Research*. **2011b**, 190(1), 339-358. DOI: https://doi.org/10.1016/j.ijpe.2011.05.015.

28. Jena, S. D.;Poggi, M. Harvest planning in the Brazilian sugar cane industry via mixed integer programming. *European Journal of Operational Research*. **2013**, 230(2), 374-384. DOI: https://doi.org/10.1016/j.ejor.2013.04.011.

29. Bohle, C.; Maturana, S.;Vera, J. A robust optimization approach to wine grape harvesting scheduling. *European Journal of Operational Research*. **2010**, 200(1), 245-252. DOI: https://doi.org/10.1016/j.ejor.2008.12.003.

30. Munhoz, J. R.;Morabito, R. Optimization approaches to support decision making in the production planning of a citrus company: A Brazilian case study. *Computers and Electronics in Agriculture*. **2014**, 107, 45-57. DOI: https://doi.org/10.1016/j.compag.2014.05.016.

31. Darby-Dowman, K.; Barker, S.; Audsley, E.;Parsons, D. A two-stage stochastic programming with recourse model for determining robust planting plans in

horticulture. *Journal of the Operational Research Society*. **2000**, 83-89. DOI: https://doi.org/10.1057/palgrave.jors.2600858.

32. Kazaz, B. Production planning under yield and demand uncertainty with yield-dependent cost and price. *Manufacturing & Service Operations Management*. **2004**, 6(3), 209-224. DOI: https://doi.org/10.1287/msom.1030.0024.

33. Ahumada, O.; Villalobos, J. R.;Mason, A. N. Tactical planning of the production and distribution of fresh agricultural products under uncertainty. *Agricultural Systems*. **2012**, 112, 17-26. DOI: https://doi.org/10.1016/j.agsy.2012.06.002.

34. González-Araya, M. C.; Soto-Silva, W. E.;Espejo, L. G. A. Harvest Planning in Apple Orchards Using an Optimization Model, in *Handbook of Operations Research in Agriculture and the Agri-Food Industry*, L.M. Plà-Aragonés, Editor, 2015, Springer: New York, NY. p. 79-105.

35. Bezat-Jarzębowska, A.;Rembisz, W. Efficiency-focused economic modeling of competitiveness in the agri-food sector. *Procedia-Social and Behavioral Sciences*. **2013**, 81, 359-365. DOI: https://doi.org/10.1016/j.sbspro.2013.06.443.

36. Ampatzidis, Y. G.; Vougioukas, S. G.; Whiting, M. D.;Zhang, Q. Applying the machine repair model to improve efficiency of harvesting fruit. *Biosystems engineering*. **2014**, 120, 25-33. DOI: https://doi.org/10.1016/j.biosystemseng.2013.07.011.

37. Miller, T. C. *Hierarchical operations and supply chain planning*. Springer Science & Business Media: London, 2012. ISBN: 978-1-4471-0305-9.

38. Birge, J. R.;Louveaux, F. *Introduction to stochastic programming*. Springer Series in Operations Research and Financial Engineering, ed. T.V. Mikosch, S.I. Resnick, and S.M. Robinson. Springer Science & Business Media, 2011. ISBN: 1461402379.

39. Nguyen, T.-D.; Nguyen-Quang, T.; Venkatadri, U.; Diallo, C.;Adams, M. Mathematical programming models for fresh fruit supply chain optimization: a review of the literature and emerging trends. *AgriEngineering*. **2021**, 3(3), 519-541. DOI: https://doi.org/10.3390/agriengineering3030034.

40. O'Byrne, R. *7 Things That Matter Most To Fresh Supply Chain Leaders*. *Logistics Bureau Pty Ltd. SEO*. 2016. Available online: https://www.logisticsbureau.com/7-things-that-matter-most-to-fresh-supply-chain-leaders/ (accessed on March 12, 2017).

41. Nguyen, T.-D.; Venkatadri, U.; Nguyen-Quang, T.; Diallo, C.;Adams, M. Optimization Model for Fresh Fruit Supply Chains: Case-Study of Dragon Fruit in Vietnam. *AgriEngineering*. **2020**, 2(1), 1-26. DOI: https://doi.org/10.3390/agriengineering2010001.

42. Humphrey, J.; McCulloch, N.;Ota, M. The impact of European market changes on employment in the Kenyan horticulture sector. *Journal of International Development*. **2004**, 16(1), 63-80. DOI: https://doi.org/10.1002/jid.1063.

43. The-Statistics-Portal, *Fruit Dossier*, 2018, The Statistics Portal.

44. Joshi, P. K.; Gulati, A.; Birthal, P. S.;Tewari, L. Agriculture diversification in South Asia: Patterns, determinants and policy implications. *Economic and political weekly*. **2004**, 2457-2467.

45. WorldBank; Organization, F. a. A.;Development, I. F. f. A. *Gender in agriculture sourcebook*. World Bank Publications: Washington, DC, USA, 2009. ISBN: 0821375873.

46. FAO, *World Food and Agriculture—Statistical Yearbook*, 2020, Rome, Italy.

47. Glen, J. J. Mathematical models in farm planning: A survey. *Operations Research*. **1987**, 35(5), 641-666.

48. Lowe, T. J.;Preckel, P. V. Decision technologies for agribusiness problems: A brief review of selected literature and a call for research. *Manufacturing & Service Operations Management*. **2004**, 6(3), 201-208.

49. Lucas, M. T.;Chhajed, D. Applications of location analysis in agriculture: a survey. *Journal of the Operational Research Society*. **2004**, 55(6), 561-578.

50. Ahumada, O.;Villalobos, J. R. Application of planning models in the agri-food supply chain: A review. *European journal of Operational research*. **2009**, 196(1), 1-20. DOI: https://doi.org/10.1016/j.ejor.2008.02.014.

51. Soto-Silva, W. E.; Nadal-Roig, E.; González-Araya, M. C.;Pla-Aragones, L. M. Operational research models applied to the fresh fruit supply chain. *European Journal of Operational Research*. **2016**, 251(2), 345-355. DOI: https://doi.org/10.1016/j.ejor.2015.08.046.

52. Agarwal, S. Issues in supply chain planning of fruits and vegetables in agri-food supply chain: A review of certain aspects. *IMS Business School, Kolkata, India*. **2017**.

53. Karp, D., *Most of America's Fruit Is Now Imported. Is That a Bad Thing?*, in *The New York Times* 2018.

54. Shukla, M.;Jharkharia, S. Agri-fresh produce supply chain management: a state-of-the-art literature review. *International Journal of Operations & Production Management*. **2013**, 33(2), 114-158. DOI: https://doi.org/10.1108/01443571311295608.

55. Willis, C.;Hanlon, W. Temporal Model for Long-Run Orchard Decisions. *Canadian Journal of Agricultural Economics/Revue canadienne d'agroeconomie*. **1976**, 24(3), 17-28. DOI: https://doi.org/10.1111/j.1744-7976.1976.tb02818.x.

56. Starbird, S. A. Optimal loading sequences for fresh-apple storage facilities. *Journal of the Operational Research Society*. **1988**, 911-917. DOI: https://doi.org/10.1057/jors.1988.157.

57. Saedt, A. P.; Hendriks, T. H.;Smits, F. M. A transition planning method applied in a decision support system for potplant nurseries. *European journal of operational research*. **1991**, 52(2), 142-154. DOI: https://doi.org/10.1016/0377-2217(91)90076-8.

58. Annevelink, E. Operational planning in horticulture: Optimal space allocation in pot-plant nurseries using heuristic techniques. *Journal of agricultural engineering research*. **1992**, 51, 167-177. DOI: https://doi.org/10.1016/0021-8634(92)80035-Q.

59. Purcell, D. L.; Turner, S. C.; Houston, J.;Hall, C. A portfolio approach to landscape plant production and marketing. *Journal of Agricultural and Applied Economics*. **1993**, 25(2), 13-26. DOI: https://doi.org/10.1017/S1074070800018927.

60. van Berlo, J. M. A decision support tool for the vegetable processing industry; an integrative approach of market, industry and agriculture. *Agricultural Systems*. **1993**, 43(1), 91-109. DOI: https://doi.org/10.1016/0308-521X(93)90094-I.

61. Miller, W. A.; Leung, L. C.; Azhar, T. M.;Sargent, S. Fuzzy production planning model for fresh tomato packing. *International Journal of Production Economics*. **1997**, 53(3), 227-238. DOI: https://doi.org/10.1016/S0925-5273(97)00110-2.

62. Stokes, J. R.; Mjelde, J. W.;Hall, C. R. Optimal marketing of nursery crops from container-based production systems. *American journal of agricultural economics*. **1997**, 79(1), 235-245. DOI: https://doi.org/10.2307/1243957.

63. Broekmeulen, R. A. C. M. Operations management of distribution centers for vegetables and fruits. *International Transactions in Operational Research*. **1998**, 5(6), 501-508. DOI: https://doi.org/10.1111/j.1475-3995.1998.tb00132.x.

64. Leutscher, K. J.; Renkema, J. A.;Challa, H. Modelling operational adaptations of tactical production plans on pot plant nurseries: a simulation approach. *Agricultural systems*. **1999**, 59(1), 67-78. DOI: https://doi.org/10.1016/S0308-521X(98)00084-5.

65. Romero, C. Risk programming for agricultural resource allocation: A multidimensional risk approach. *Annals of Operations Research*. **2000**, 94(1), 57-68. DOI: https://doi.org/10.1023/A:1018985620677.

66. Gigler, J. K.; Hendrix, E. M. T.; Heesen, R. A.; van den Hazelkamp, V. G. W.;Meerdink, G. On optimisation of agri chains by dynamic programming. *European Journal of Operational Research*. **2002**, 139(3), 613-625. DOI: https://doi.org/10.1016/S0377-2217(01)00191-6.

67. Hester, S. M.;Cacho, O. Modelling apple orchard systems. *Agricultural systems*. **2003**, 77(2), 137-154. DOI: https://doi.org/10.1016/S0308-521X(02)00106-3.

68. Itoh, T.; Ishii, H.;Nanseki, T. A model of crop planning under uncertainty in agricultural management. *International Journal of Production Economics*. **2003**, 81, 555-558. DOI: https://doi.org/10.1016/S0925-5273(02)00283-9.

69. Vitoriano, B.; Ortuño, M. T.; Recio, B.; Rubio, F.;Alonso-Ayuso, A. Two alternative models for farm management: discrete versus continuous time horizon. *European Journal of Operational Research*. **2003**, 144(3), 613-628. DOI: https://doi.org/10.1016/S0377-2217(02)00143-1.

70. Allen, S. J.;Schuster, E. W. Controlling the risk for an agricultural harvest. *Manufacturing & Service Operations Management*. **2004**, 6(3), 225-236. DOI: https://doi.org/10.1287/msom.1040.0035.

71. Rantala, J. Optimizing the supply chain strategy of a multi-unit Finnish nursery company. **2004**. DOI: https://doi.org/10.14214/sf.429.

72. Blanco, A. M.; Masini, G.; Petracci, N.;Bandoni, J. A. Operations management of a packaging plant in the fruit industry. *Journal of Food Engineering*. **2005**, 70(3), 299-307. DOI: https://doi.org/10.1016/j.jfoodeng.2004.05.075.

73. Ortmann, F. G.; Van Vuuren, J.;Van Dyk, F. Modelling the South African fruit export infrastructure: A case study. *ORiON*. **2006**, 22(1), 35-57. DOI: https://doi.org/10.5784/22-1-32.

74. Bai, R.; Burke, E. K.;Kendall, G. Heuristic, meta-heuristic and hyper-heuristic approaches for fresh produce inventory control and shelf space allocation. *Journal of the Operational Research Society*. **2008**, 59(10), 1387-1397. DOI: https://doi.org/10.1057/palgrave.jors.2602463.

75. Cittadini, E. D.; Lubbers, M. T. M. H.; de Ridder, N.; Van Keulen, H.;Claassen, G. D. H. Exploring options for farm-level strategic and tactical decision-making in fruit production systems of South Patagonia, Argentina. *Agricultural Systems*. **2008**, 98(3), 189-198. DOI: https://doi.org/10.1016/j.agsy.2008.07.001.

76. Blackburn, J.;Scudder, G. Supply chain strategies for perishable products: the case of fresh produce. *Production and Operations Management*. **2009**, 18(2), 129-137. DOI: https://doi.org/10.1111/j.1937-5956.2009.01016.x.

77. Van Der Vorst, J. G.; Tromp, S.-O.;Zee, D.-J. v. d. Simulation modelling for food supply chain redesign; integrated decision making on product quality, sustainability

and logistics. *International Journal of Production Research*. **2009**, 47(23), 6611-6631. DOI: https://doi.org/10.1080/00207540802356747.

78. Arnaout, J. P. M.;Maatouk, M. Optimization of quality and operational costs through improved scheduling of harvest operations. *International Transactions in Operational Research*. **2010**, 17(5), 595-605. DOI: https://doi.org/10.1111/j.1475-3995.2009.00740.x.

79. Morande, H.;Maturana, S. Design and validation of a decision support system for oenologists. *Revista ICHIO*. **2010**, 1(1), 46-58.

80. Arumugam, N.; Fatimah, M. A.; Chiew, E. F.;Zainalabidin, M. Supply chain analysis of fresh fruits and vegetables (FFV): prospects of contract farming. *Agricultural Economics (Zemědělská Ekonomika)*. **2010**, 56(9), 435-442. DOI: https://doi.org/10.17221/30/2009-AGRICECON.

81. Verdouw, C.; Beulens, A.; Trienekens, J.;Wolfert, J. Process modelling in demand-driven supply chains: A reference model for the fruit industry. *Computers and electronics in agriculture*. **2010**, 73(2), 174-187. DOI: https://doi.org/10.1016/j.compag.2010.05.005.

82. Jang, W.;Klein, C. M. Supply chain models for small agricultural enterprises. *Annals of Operations Research*. **2011**, 190(1), 359-374. DOI: https://doi.org/10.1007/s10479-009-0521-8.

83. Jia, X.;Huang, J. Contractual arrangements between farmer cooperatives and buyers in China. *Food Policy*. **2011**, 36(5), 656-666. DOI: https://doi.org/10.1016/j.foodpol.2011.06.007.

84. Rong, A.; Akkerman, R.;Grunow, M. An optimization approach for managing fresh food quality throughout the supply chain. *International Journal of Production Economics*. **2011**, 131(1), 421-429. DOI: https://doi.org/10.1016/j.ijpe.2009.11.026.

85. Amorim, P.; Günther, H.-O.;Almada-Lobo, B. Multi-objective integrated production and distribution planning of perishable products. *International Journal of Production Economics*. **2012**, 138(1), 89-101. DOI: https://doi.org/10.1016/j.ijpe.2012.03.005.

86. Banaeian, N.; Omid, M.;Ahmadi, H. Greenhouse strawberry production in Iran, efficient or inefficient in energy. *Energy Efficiency*. **2012**, 5(2), 201-209. DOI: http://dx.doi.org/10.1007/s12053-011-9133-7.

87. Perdana, T. The Triple Helix Model for Fruits and Vegetables Supply Chain Management Development Involving Small Farmers in Order to Fulfill the Global Market Demand: A Case Study in "Value Chain Center (VCC) Universitas Padjadjaran". *Procedia-Social and Behavioral Sciences*. **2012**, 52, 80-89. DOI: https://doi.org/10.1016/j.sbspro.2012.09.444.

88. Yu, Y.; Wang, Z.;Liang, L. A vendor managed inventory supply chain with deteriorating raw materials and products. *International Journal of Production Economics*. **2012**, 136(2), 266-274. DOI: https://doi.org/10.1016/j.ijpe.2011.11.029.

89. Catalá, L. P.; Durand, G. A.; Blanco, A. M.;Bandoni, J. A. Mathematical model for strategic planning optimization in the pome fruit industry. *Agricultural Systems*. **2013**, 115, 63-71. DOI: https://doi.org/10.1016/j.agsy.2012.09.010.

90. Lambert, G. F.; Lasserre, A. A. A.; Ackerman, M. M.; Sánchez, C. G. M.; Rivera, B. O. I.;Azzaro-Pantel, C. An expert system for predicting orchard yield and fruit quality and its impact on the Persian lime supply chain. *Engineering Applications of Artificial Intelligence*. **2014**, 33, 21-30. DOI: https://doi.org/10.1016/j.engappai.2014.03.013.

91. Rocco, C. D.;Morabito, R. Scheduling of production and logistics operations of steam production systems in food industries: a case study of the tomato processing industry. *Journal of the Operational Research Society*. **2014**, 65(12), 1896-1904. DOI: https://doi.org/10.1057/jors.2013.84.

92. Velychko, O. Integrated modeling of solutions in the system of distributing logistics of a fruit and vegetable cooperative. *Business: Theory and Practice/Verslas: Teorija ir Praktika*. **2014**, 15(4), 362-370.

93. Nadal-Roig, E.;Plà-Aragonés, L. M. Optimal Transport Planning for the Supply to a Fruit Logistic Centre, in *Handbook of Operations Research in Agriculture and the Agri-Food Industry*, 2015, Springer: New York, NY, USA. p. 163-177.

94. Catalá, L. P.; Moreno, M. S.; Blanco, A. M.;Bandoni, J. A. A bi-objective optimization model for tactical planning in the pome fruit industry supply chain. *Computers and Electronics in Agriculture*. **2016**, 130, 128-141. DOI: https://doi.org/10.1016/j.compag.2016.10.008.

95. Rocco, C. D.;Morabito, R. Production and logistics planning in the tomato processing industry: A conceptual scheme and mathematical model. *Computers and Electronics in Agriculture*. **2016**, 127, 763-774. DOI: https://doi.org/10.1016/j.compag.2016.08.002.

96. Grillo, H.; Alemany, M. M. E.; Ortiz, A.;Fuertes-Miquel, V. S. Mathematical modelling of the order-promising process for fruit supply chains considering the perishability and subtypes of products. *Applied Mathematical Modelling*. **2017**, 49, 255-278. DOI: https://doi.org/10.1016/j.apm.2017.04.037.

97. Soto-Silva, W. E.; González-Araya, M. C.; Oliva-Fernández, M. A.;Plà-Aragonés, L. M. Optimizing fresh food logistics for processing: Application for a large Chilean apple supply chain. *Computers and Electronics in Agriculture*. **2017**, 136, 42-57. DOI: https://doi.org/10.1016/j.compag.2017.02.020.

98. Cheraghalipour, A.; Paydar, M. M.;Hajiaghaei-Keshteli, M. Designing and solving a bi-level model for rice supply chain using the evolutionary algorithms. *Computers and Electronics in Agriculture*. **2019**, 162, 651-668. DOI: https://doi.org/10.1016/j.compag.2019.04.041.

99. Foong, S. Z.; Goh, C. K.; Supramaniam, C. V.;Ng, D. K. Input–output optimisation model for sustainable oil palm plantation development. *Sustainable Production and Consumption*. **2019**, 17, 31-46. DOI: https://doi.org/10.1016/j.spc.2018.08.010.

100. Gokarn, S.;Kuthambalayan, T. S. Creating sustainable fresh produce supply chains by managing uncertainties. *Journal of cleaner production*. **2019**, 207, 908-919. DOI: https://doi.org/10.1016/j.jclepro.2018.10.072.

101. Ji, Y.; Du, J.; Han, X.; Wu, X.; Huang, R.; Wang, S.;Liu, Z. A mixed integer robust programming model for two-echelon inventory routing problem of perishable products. *Physica A: Statistical Mechanics and its Applications*. **2020**, 548, 124481. DOI: https://doi.org/10.1016/j.physa.2020.124481.

102. Varas, M.; Basso, F.; Maturana, S.; Osorio, D.;Pezoa, R. A multi-objective approach for supporting wine grape harvest operations. *Computers & Industrial Engineering*. **2020**, 145, 106497. DOI: https://doi.org/10.1016/j.cie.2020.106497.

103. Alemany, M. M. E.; Esteso, A.; Ortiz, Á.;del Pino, M. Centralized and distributed optimization models for the multi-farmer crop planning problem under uncertainty: Application to a fresh tomato Argentinean supply chain case study. *Computers & Industrial Engineering*. **2021**, 153, 107048. DOI: https://doi.org/10.1016/j.cie.2020.107048.

104. Gómez-Lagos, J. E.; González-Araya, M. C.; Soto-Silva, W. E.;Rivera-Moraga, M. M. Optimizing tactical harvest planning for multiple fruit orchards using a metaheuristic modeling approach. *European journal of operational research*. **2021**, 290(1), 297-312. DOI: https://doi.org/10.1016/j.ejor.2020.08.015.

105. Ktenioudaki, A.; O'Donnell, C. P.; Emond, J. P.;do Nascimento Nunes, M. C. Blueberry supply chain: Critical steps impacting fruit quality and application of a boosted regression tree model to predict weight loss. *Postharvest Biology and Technology*. **2021**, 179, 111590. DOI: https://doi.org/10.1016/j.postharvbio.2021.111590.

106. Lim, C. H.; Cheah, Z. H.; Lee, X. H.; How, B. S.; Ng, W. P. Q.; Ngan, S. L.; Lim, S.;Lam, H. L. Harvesting and evacuation route optimisation model for fresh fruit bunch in the oil palm plantation site. *Journal of Cleaner Production*. **2021**, 307, 127238. DOI: https://doi.org/10.1016/j.jclepro.2021.127238.

107. Trivedi, A.; Sohal, A.; Joshi, S.;Sharma, M. A two-stage optimization model for tactical planning in fresh fruit supply chains: A case study of Kullu, India. *International Journal of Supply and Operations Management*. **2021**, 8(1), 18-28. DOI: https://dx.doi.org/10.22034/IJSOM.2021.1.2.

108. Diop, N.;Jaffee, S. M. Fruits and vegetables: global trade and competition in fresh and processed product markets, in *Global Agricultural Trade and Developing Countries*, M.A. Aksoy and J.C. Beghin, Editors, 2005, World Bank: Washington, DC, USA. p. 237-257.

109. Audsley, E.;Sandars, D. L. A review of the practice and achievements from 50 years of applying OR to agricultural systems in Britain. *OR Insight*. **2009**, 22(1), 2-18.

110. Zimmermann, H.-J. Description and optimization of fuzzy systems. *International Journal of General System*. **1975**, 2(1), 209-215. DOI: https://doi.org/10.1080/03081077508960870.

111. Cobb, C. W.;Douglas, P. H. A theory of production. *The American Economic Review*. **1928**, 18(1), 139-165.

112. Monostori, J. Supply chains robustness: Challenges and opportunities. in *Procedia CIRP*, 2018.

113. Ten Berge, H. F. M.; Van Ittersum, M. K.; Rossing, W. A. H.; Van de Ven, G. W. J.;Schans, J. Farming options for The Netherlands explored by multi-objective modelling. *European Journal of Agronomy*. **2000**, 13(2), 263-277. DOI: https://doi.org/10.1016/S1161-0301(00)00078-2.

114. Nielsen, I. Grubben, G. J. H. & Denton, O. A. (eds) 2004. Plant resources of Tropical Africa 2. *Nordic Journal of Botany*. **2003**, 23, 298. DOI: https://doi.org/10.1111/J.1756-1051.2003.TB00397.X.

115. Defraeye, T.; Nicolai, B.; Kirkman, W.; Moore, S.; van Niekerk, S.; Verboven, P.;Cronjé, P. Integral performance evaluation of the fresh-produce cold chain: A case study for ambient loading of citrus in refrigerated containers. *Postharvest Biology and Technology*. **2016**, 112, 1-13.

116. Wu, W.;Defraeye, T. Identifying heterogeneities in cooling and quality evolution for a pallet of packed fresh fruit by using virtual cold chains. *Applied Thermal Engineering*. **2018**, 133, 407-417.

117. Raut, R. D.; Gardas, B. B.; Narwane, V. S.;Narkhede, B. E. Improvement in the food losses in fruits and vegetable supply chain-a perspective of cold third-party logistics approach. *Operations research perspectives*. **2019**, 6, 100117. DOI: https://doi.org/10.1016/j.orp.2019.100117.

118. Wu, W.; Beretta, C.; Cronje, P.; Hellweg, S.;Defraeye, T. Environmental trade-offs in fresh-fruit cold chains by combining virtual cold chains with life cycle assessment. *Applied Energy*. **2019**, 254, 113586.

119. Zhao, X.; Xia, M.; Wei, X.; Xu, C.; Luo, Z.;Mao, L. Consolidated cold and modified atmosphere package system for fresh strawberry supply chains. *LWT*. **2019**, 109, 207-215.

120. Luong, N. T. L.;Nguyen, M. C. Increasing Market Access of Selected Tropical Fruit Through Value Chain Improvements in Vietnam. *Food and Fertilizer Technology Center*. Taipei, Taiwan. 2015. http://www.fftc.agnet.org/library.php?func=view&id=20150805092541&type_id=4 (accessed on July 09, 2017).

121. AXIS-Research. *Report on Value Chain of Dragon Fruit of Binh Thuan Province*. *Information Center for Agriculture and Rural development, Vietnam (Agroinfo)*. Vietnam. 2009. http://agro.gov.vn/images/2007/05/Dragon_Fruit_in_BT(E).pdf. (accessed on May 21, 2015).

122. Kiet; Cuoc;Manh. Binh Thuan, Vietnam. Personal Communication, 2017.

123. Ben-Tal, A.; El Ghaoui, L.;Nemirovski, A. *Robust optimization*. Robust optimization. Princeton University Press, 2009. ISBN: 1400831059.

124. Nguyen, T.-D.; Venkatadri, U.; Nguyen-Quang, T.; Diallo, C.; Pham, D.-H.; Phan, H.-T.; Pham, L.-K.; Nguyen, P.-C.;Adams, M. Stochastic Modelling Frameworks for Dragon Fruit Supply Chains in Vietnam under Uncertain Factors. *Sustainability*. **2024**, 16(6), 2423. DOI: https://doi.org/10.3390/su16062423.

125. FAO; WFP; UNECE; UNICEF; WHO;WMO. *Regional Overview of Food Security and Nutrition in Europe and Central Asia 2020: Affordable Healthy Diets to Address All Forms of Malnutrition for Better Health*. Europe and Central Asia—Regional Overview of Food Security and, Nutrition. Vol. 2020. Budapest, Hungary, 2021.

126. Bochtis, D.; Benos, L.; Lampridi, M.; Marinoudi, V.; Pearson, S.;Sørensen, C. G. Agricultural workforce crisis in light of the COVID-19 pandemic. *Sustainability*. **2020**, 12(19), 8212. DOI: https://doi.org/10.3390/su12198212.

127. OECD. Connecting Businesses and Consumers during COVID-19: Trade in Parcels. *OECD Publishing*. Paris, France. 2020. https://www.oecd.org/coronavirus/policy-responses/connecting-businesses-and-consumers-during-covid-19-tradein-parcels-d18de131/#biblio-d1e1182 (accessed on 28 February 2023).

128. Le, T. H. X. *Báo Cáo Thị Trường Gạo Tháng 11 Năm 2021*. Ho Chi Minh City, Vietnam,. 2021. (accessed on

129. MOIT. *Vietnam Import—Export Report 2020*. Hanoi, Vietnam. 2021. (accessed on

130. Trinh, N. T. T. Impact of the COVID-19 on the labor market in Vietnam. *International Journal of Health Sciences*. **2022**, 6(S1), 8355–8367. DOI: https://doi.org/10.53730/ijhs.v6nS1.6973.

131. Nguyen, T.-D.; Nguyen-Quang, T.; Venkatadri, U.; Diallo, C.;Adams, M. Mathematical programming models for fresh fruit supply chain optimization: a review of the literature and emerging trends. *AgriEngineering*. **2021**, 3(3), 519-541. DOI: https://doi.org/10.3390/agriengineering3030034.

132. Nguyen, H. Giải pháp thúc đẩy tiêu thụ nông sản trong mùa dịch Corona. *Ministry of Finance, Vietnam*. Hanoi, Vietnam. 2020. https://mof.gov.vn/webcenter/portal/vclvcstc/pages_r/l/chi-tiet-tin?dDocName=MOFUCM171933 (accessed on 28 February 2022).

133. Van-Anh. Thanh Long Vàng Giá 700.000 d¯ồng/kg Vẫn Hút Khách Nhà Giàu. 2016. https://eva.vn/bep-eva/thanh-long-vang-gia-700000-dong-kg-van-hut-khach-nha-giau-c162a286921.html (accessed on 28 February 2022).

134. Van_Anh. Thanh long vàng giá 700.000 đồng/kg vẫn hút khách nhà giàu. 2016. Available online: https://eva.vn/bep-eva/thanh-long-vang-gia-700000-dong-kg-van-hut-khach-nha-giau-c162a286921.html (accessed on 28 February 2023).

135. Minh_Dam;Trong_Linh. Trồng thử thanh long vỏ vàng. *Báo Nông nghiệp Việt Nam*. Hanoi, Vietnam. 2020. https://nongnghiep.vn/trong-thu-thanh-long-vo-vang-d264576.html (accessed on 28 February 2023).

136. Eckstein, D.; Künzel, V.; Schäfer, L.;Winges, M. *Global Climate Rate Index 2020: Who suffers most from extreme weather events? Weather-related loss events in 2018 and 1999 to 2018*. Germanwatch e.V.: Berlin, Germany, 2019. ISBN: 978-3-943704-77-8.

137. ADB. *Climate Change Risk and Adaptation Assessment for Irrigation in Southern Viet Nam: Water Efficiency Improvements in Drought-Affected Provinces*. Manila, Philippines. 2020. (accessed on 28 February 2023).

138. Hai, L. T.; Gobin, A.;Hens, L. Risk assessment of desertification for Binh Thuan province, Vietnam. *Human and Ecological Risk Assessment: An International Journal*. **2013**, 19(4), 1544-1556. DOI: https://doi.org/10.1080/10807039.2012.716688.

139. Gobin, A.; Le Trinh, H.; Pham Ha, L.;Hens, L. *Land use and desertification in the Binh Thuan Province of Southeastern Vietnam: mitigation and adaptation options now and under climate change* in European Geosciences Union General Assembly 2012, 2012. Vienna, Austria. p. 8398.

140. *Climate Change Phan Thiết. University of Basel*. Basel, Switzerland. 2021. Available online:

https://www.meteoblue.com/en/weather/historyclimate/change/phan-thiết_vietnam_1571058 (accessed on 28 February 2023).

141. Le, T.; Sun, C.; Choy, S.;Kuleshov, Y. Regional drought risk assessment in the Central Highlands and the South of Vietnam. *Geomatics, Natural Hazards and Risk*. **2021**, 12(1), 3140-3159. DOI: https://doi.org/10.1080/19475705.2021.1998232.

142. Doutreloup, S.; Erpicum, M.; Fettweis, X.;Ozer, P. Analysis of the past (1970-1999) and future (2046-2065 and 2081-2100) evolutions of precipitation and temperature, in the Province of Binh Thuan, South East Vietnam, based on IPCC models. in *1st International Conference on Energy, Environment and Climate Change*, 2011. Ho Chi Minh City, Vietnam.

143. Dang, T. B.;Phan, H. Mối quan hệ "Nước và biến đổi khí hậu" - "Khí hậu và nước" tại Ninh Thuận. 2020. Available online: http://vmha.gov.vn/public/index.php/tin-tuc-bdkh-112/moi-quan-he-"nuoc-va-bien-doi-khi-hau"-"khi-hau-va-nuoc"-tai-ninh-thuan-6103.html (accessed on 28 February 2023).

144. Vinh, P. Q.;Hương, P. T. T. Assessing agricultural drought for Binh Thuan province under climate change scenario. *Vietnam Journal of Earth Sciences*. **2012**, 34(4), 513-523.

145. Tran, T.; Nguyen, V.; Huynh, T.; Mai, V.; Nguyen, X.;Doan, H. *Scenarios of climate change and sea level rise for Vietnam*. Hanoi, Vietnam. 2016. (accessed on

146. Van Hong, N.;Nguyen, V. T. The impact of Climate Change on the transportation in Binh Thuan Province. *Vietnam Journal of Hydrometeorology*. **2021**, 8, 9-15. DOI: https://dx.doi.org/10.36335/vnjhm.2021(8).9-15.

147. Agnolucci, P.; Rapti, C.; Alexander, P.; De Lipsis, V.; Holland, R. A.; Eigenbrod, F.;Ekins, P. Impacts of rising temperatures and farm management practices on global yields of 18 crops. *Nature Food*. **2020**, 1(9), 562-571. DOI: https://doi.org/10.1038/s43016-020-00148-x.

148. USEPA. Climate impacts on agriculture and food supply. 2016. Available online: https://climatechange.chicago.gov/climate-impacts/climate-impacts-agriculture-and-food-supply#:~:text=Climate%20change%20can%20disrupt%20food,result%20in%20reduced%20agricultural%20productivity (accessed on 28 February 2023).

149. Knight, F. H. *Risk, uncertainty and profit* Vol. 31. Houghton Mifflin Company: Boston and New York, 1921.

150. Carøe, C. C.;Schultz, R. Dual decomposition in stochastic integer programming. *Operations Research Letters*. **1999**, 24(1-2), 37-45. DOI: https://doi.org/10.1016/S0167-6377(98)00050-9.

151. Pourmohammadi, F.; Teimoury, E.;Gholamian, M. A scenario-based stochastic programming approach for designing and planning wheat supply chain (A case study). *Decision Science Letters*. **2020**, 9(4), 537-546. DOI: http://dx.doi.org/10.5267/j.dsl.2020.8.004.

152. Jacquet, F.;Pluvinage, J. Climatic uncertainty and farm policy: a discrete stochastic programming model for cereal-livestock farms in Glgeria. *Agricultural systems*. **1997**, 53(4), 387-407. DOI: https://doi.org/10.1016/0308-521X(95)00076-H.

153. Tan, B.;Çömden, N. Agricultural planning of annual plants under demand, maturation, harvest, and yield risk. *European Journal of Operational Research*. **2012**, 220(2), 539-549. DOI: https://doi.org/10.1016/j.ejor.2012.02.005.

154. Costa, A. M.; dos Santos, L. M. R.; Alem, D. J.;Santos, R. H. S. Sustainable vegetable crop supply problem with perishable stocks. *Annals of Operations Research*. **2014**, 219, 265-283. DOI: https://doi.org/10.1007/s10479-010-0830-y.

155. Marchal, P. C.; Gila, D. M.; García, J. G.;Ortega, J. G. Stochastic season-wide optimal production planning of virgin olive oil. *Journal of Process Control*. **2018**, 72, 64-73. DOI: https://doi.org/10.1016/j.jprocont.2018.08.001.

156. Flores, H.;Villalobos, J. R. A stochastic planning framework for the discovery of complementary, agricultural systems. *European Journal of Operational Research*. **2020**, 280(2), 707-729. DOI: https://doi.org/10.1016/j.ejor.2019.07.053.

157. An, K.;Ouyang, Y. Robust grain supply chain design considering post-harvest loss and harvest timing equilibrium. *Transportation Research Part E: Logistics and Transportation Review*. **2016**, 88, 110-128. DOI: https://doi.org/10.1016/j.tre.2016.01.009.

158. Banasik, A.; Kanellopoulos, A.; Bloemhof-Ruwaard, J. M.;Claassen, G. D. H. Accounting for uncertainty in eco-efficient agri-food supply chains: A case study for mushroom production planning. *Journal of cleaner production*. **2019**, 216, 249-256. DOI: https://doi.org/10.1016/j.jclepro.2019.01.153.

159. Chavez, M. M. M.; Sarache, W.; Costa, Y.;Soto, J. Multiobjective stochastic scheduling of upstream operations in a sustainable sugarcane supply chain. *Journal of Cleaner Production*. **2020**, 276, 123305. DOI: https://doi.org/10.1016/j.jclepro.2020.123305.

160. IPCC. *Climate Change 2014: Impacts, Adaptation, and Vulnerability. Part A: Global and Sectoral Aspects*. Contribution of Working Group II to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge University Press: Cambridge, UK,, 2014. 1132. ISBN: 1107058163.

161. Dai, A. Increasing drought under global warming in observations and models. *Nature climate change*. **2013**, 3(1), 52-58. DOI: https://doi.org/10.1038/nclimate1633.

162. Lesk, C.; Rowhani, P.;Ramankutty, N. Influence of extreme weather disasters on global crop production. *Nature*. **2016**, 529(7584), 84-87. DOI: https://doi.org/10.1038/nature16467.

163. Duangdai, E.;Likasiri, C. Mathematical model analyses on the effects of global temperature and forest cover on seasonal rainfalls: A Northern Thailand case study. *Journal of Hydrology*. **2015**, 524, 270-278. DOI: https://doi.org/10.1016/j.jhydrol.2015.02.043.

164. Sun, G.-Q.; Li, L.; Li, J.; Liu, C.; Wu, Y.-P.; Gao, S.; Wang, Z.;Feng, G.-L. Impacts of climate change on vegetation pattern: Mathematical modelling and data analysis. *Physics of Life Reviews*. **2022**. DOI: https://doi.org/10.1016/j.plrev.2022.09.005.

165. Ghaffari, A.; Nasseri, M.;Someeh, A. P. Assessing the economic effects of drought using Positive Mathematical Planning model under climate change scenarios. *Heliyon*. **2022**, 8(12). DOI: https://doi.org/10.1016/j.heliyon.2022.e11941.

166. Kung, C.-C.;Wu, T. Influence of water allocation on bioenergy production under climate change: A stochastic mathematical programming approach. *Energy*. **2021**, 231, 120955. DOI: https://doi.org/10.1016/j.energy.2021.120955.

167. Raiffa, H. Decision analysis: introductory lectures on choices under uncertainty. **1968**. DOI: https://doi.org/10.2307/2987280.

168. Dantzig, G. B.;Infanger, G. Large-scale stochastic linear programs: Importance sampling and Benders decomposition. in *IMACS*, 1991.

169. Calfa, B. A.; Agarwal, A.; Grossmann, I. E.;Wassick, J. M. Data-driven multi-stage scenario tree generation via statistical property and distribution matching. *Computers & Chemical Engineering*. **2014**, 68, 7-23. DOI: https://doi.org/10.1016/j.compchemeng.2014.04.012.

170. Kim, S.; Pasupathy, R.;Henderson, S. G. A guide to sample average approximation, in *Handbook of simulation optimization*, M. Fu, Editor, 2015, Springer: New York, NY. p. 207-243.

171. Wang, Y.; Pan, B.; Tu, W.; Liu, P.; Jiang, B.; Gao, C.; Lu, W.; Jui, S.;Kong, L. Sample average approximation for stochastic optimization with dependent data: Performance guarantees and tractability. in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

172. Verweij, B.; Ahmed, S.; Kleywegt, A. J.; Nemhauser, G.;Shapiro, A. The sample average approximation method applied to stochastic routing problems: a computational study. *Computational optimization and applications*. **2003**, 24, 289-333. DOI: https://doi.org/10.1023/A:1021814225969.

173. Prékopa, A. *Stochastic programming*. Mathematics and Its Applications. Vol. 324. Springer Science & Business Media, 1995. ISBN: 9401730873.

174. Markowitz, H. Portfolio Selection. *The Journal of Finance*. **1952**, 7(1), 77–91. DOI: https://doi.org/10.2307/2975974.

175. Charnes, A.;Cooper, W. W. Chance-constrained programming. *Management science*. **1959**, 6(1), 73-79. DOI: https://doi.org/10.1287/mnsc.6.1.73.

APPENDIX A1 COPYRIGHT PERMISSION LETTER

May 21, 2024

AgriEngineering
MDPI, St. Alban-Anlage 66, 4052 Basel, Switzerland

I am preparing my PhD thesis for submission to the Faculty of Graduate Studies at Dalhousie University, Halifax, Nova Scotia, Canada. I am seeking your permission to include a manuscript version of the following paper(s) as a chapter in the thesis:

Nguyen, T.-D.; Venkatadri, U.; Nguyen-Quang, T.; Diallo, C.;Adams, M. Optimization Model for Fresh Fruit Supply Chains: Case-Study of Dragon Fruit in Vietnam. *AgriEngineering*. 2020, 2(1), 1-26. DOI: https://doi.org/10.3390/agriengineering2010001.

Nguyen, T.-D.; Venkatadri, U.; Nguyen-Quang, T.; Diallo, C.;Adams, M. Mathematical programming models for fresh fruit supply chain optimization: a review of the literature and emerging trends. *AgriEngineering*. 2021, 3(3), 519-541. DOI: https://doi.org/10.3390/agriengineering3030034.

Canadian graduate theses are reproduced by the Library and Archives of Canada (formerly National Library of Canada) through a nonexclusive, world-wide license to reproduce, loan, distribute, or sell theses. I am also seeking your permission for the material described above to be reproduced and distributed by the LAC(NLC). Further details about the LAC(NLC) thesis program are available on the LAC(NLC) website (www.nlc-bnc.ca).

Full publication details and a copy of this permission letter will be included in the thesis.

Yours sincerely,

Tri-Dung Nguyen

Permission is granted for:
a) the inclusion of the material described above in your thesis.
b) for the material described above to be included in the copy of your thesis that is sent to the Library and Archives of Canada (formerly National Library of Canada) for reproduction and distribution.
Name: *Dr. Mathew G. Pelletier* Title: *Editor-in-Chief*

Signature: Date: May 22, 2024

APPENDIX A2 COPYRIGHT PERMISSION LETTER

May 30, 2024

Sustainability
MDPI, St. Alban-Anlage 66, 4052 Basel, Switzerland

I am preparing my PhD thesis for submission to the Faculty of Graduate Studies at Dalhousie University, Halifax, Nova Scotia, Canada. I am seeking your permission to include a manuscript version of the following paper(s) as a chapter in the thesis:

Nguyen, T.-D.; Venkatadri, U.; Nguyen-Quang, T.; Diallo, C.; Pham, D.-H.; Phan, H.-T.; Pham, L.-K.; Nguyen, P.-C.;Adams, M. Stochastic Modelling Frameworks for Dragon Fruit Supply Chains in Vietnam under Uncertain Factors. *Sustainability*. 2024, 16(6), 2423. DOI: https://doi.org/10.3390/su16062423.

Canadian graduate theses are reproduced by the Library and Archives of Canada (formerly National Library of Canada) through a nonexclusive, world-wide license to reproduce, loan, distribute, or sell theses. I am also seeking your permission for the material described above to be reproduced and distributed by the LAC(NLC). Further details about the LAC(NLC) thesis program are available on the LAC(NLC) website (www.nlc-bnc.ca).

Full publication details and a copy of this permission letter will be included in the thesis.

Yours sincerely,

Tri-Dung Nguyen

Permission is granted for:
a) the inclusion of the material described above in your thesis.
b) for the material described above to be included in the copy of your thesis that is sent to the Library and Archives of Canada (formerly National Library of Canada) for reproduction and distribution.
Name: *Prof. Dr. Marc A. Rosen* Title: *Editor-in-Chief*

Signature: Date: May 30, 2024

APPENDIX A3 RESUME

Les fruits du dragon vietnamiens sont au premier rang du marché international, représentant plus de 50% de la production totale. Tandis que seulement de 10% à 20% de la production nationale de fruits du dragon soient consommés dans le marché domestique, de 80 à 85% sont exportés. L'Asie, l'Europe et les Amériques constituent les principaux marchés de consommation des fruits du dragon à l'échelle internationale, sur laquelle la Chine contribue à hauteur de 80 à 90 % à la valeur annuelle globale des exportations.

Néanmoins, la production et la distribution des fruits du dragon sont confrontées aux diverses incertitudes et risques, notamment les conditions météorologiques, ainsi que d'autres problèmes incluant ravageurs, maladies, fluctuations des prix du marché, défis liés à la logistique et aux infrastructures, concurrence, hausse des coûts des intrants, conditions de durabilité et règlements d'exportation. Cette thèse de recherche doctorale vise à développer les modèles mathématiques afin d'analyser et d'améliorer les approches de quantification utilisées dans les processus décisionnels complexes concernant la cultivation, la récolte et la distribution du fruit du dragon au Vietnam. En incorporant des modèles à la fois déterministes et stochastiques, nos approches de modélisation fournissent à la fois des stratégies méthodiques et centrées sur les données du terrain pour surmonter ces obstacles. En conséquence, les décideurs sont en mesure de prendre leurs décisions éclairées, ce qui à leur tour devraient améliorer l'efficacité opérationnelle, minimiser les dépenses et maximiser les performances globales dans la gestion de la chaîne d'approvisionnement des fruits du dragon.

Les processus de prise de décision sont structurés en deux étapes: 1) la pré-plantation en tenant compte les coûts et les ressources sont considérés comme déterministes; 2) après la récolte, lorsque les paramètres stochastiques devenant apparents. Les processus de prise de décision sont divisés en deux phases, tactiques et opérationnelles, en utilisant l'approche de planification hiérarchique, avantageuse pour les cultivateurs, les producteurs, les distributeurs et les vendeurs.

La thèse a finalement atteint ses trois objectifs de recherche: 1) d'avoir mené une revue complète et mise à jour de la littérature sur les modèles mathématiques concernant la production et la distribution de fruits frais ; 2) d'avoir développé un modèle d'optimisation déterministe avec certaines données du terrain ; 3) et d'avoir conçu un modèle

d'optimisation stochastique pour aborder le couplage des effets complexes et des facteurs incertains dans la production et l’exportation des fruits du dragon au Vietnam.

APPENDIX B1 SUMMARY OF SCENARIOS WITH THEIR CHARACTERISTICS.

Table B1 Summary of scenarios with their characteristics.

Scenario	Situation (Sub-Scenario or Case)	Limit of Planting Area for Each Crop	Descriptions
Baseline scenario		No limit for each crop	Demands and prices unchanged within 10 years
Changes in price of Crop 2	1	No limit for each crop	The price increasing gradually within 10 years
	2	No limit for each crop	The price decreasing gradually within 10 years
Changes in demands	1	No limit for each crop	Demand of Crop 3 increasing 4 times
	2	No limit for each crop	Demands of all crops increasing 20%
	3	No limit for each crop	Demands of all crops increasing 40%
	4	No limit for each crop	Demands of all crops increasing 80%
Crop 3 selling price with probability factor	1	No limit for each crop	0.2 for $1, 0.2 for $5, and 0.6 for $10
	2	No limit for each crop	0.2 for $1, 0.6 for $5, and 0.2 for $10
	3	No limit for each crop	0.6 for $1, 0.2 for $5, and 0.2 for $10
Land restriction		50% for Crop 1, 35% for Crop 2, 15% for Crop 3	Demands and prices unchanged within 10 years
Influence of initial land	Crop 1	No limit for each crop	All initial land used for Crop 1. Demands and prices unchanged within 10 years
	Crop 2	No limit for each crop	All initial land used for Crop 2. Demands and prices unchanged within 10 years
	Crop 3	No limit for each crop	All initial land used for Crop 2. Demands and prices unchanged within 10 years
	Crop 1—Age 1	No limit for each crop	All initial land used for Crop 1 at age 1. Demands and prices unchanged within 10 years
	Crop 1—Age 3	No limit for each crop	All initial land used for Crop 1 at age 3. Demands and prices unchanged within 10 years
	Crop 1—Age 5	No limit for each crop	All initial land used for Crop 1 at age 5. Demands and prices unchanged within 10 years

APPENDIX B2 THE MATHEMATICAL MODEL OF DISTRIBUTION MATCHING PROBLEM

Indices:

Different species of crop	j
The outcomes (branches) from the root node	o

Variables:

Probabilities of outcomes o	$prob_o$
Positive and negative variances of crop j calculated from the tree	var_j^+, var_j^-
Positive and negative co-variances of crop j and j' calculated from the tree	$cov_{jj'}^+$, $cov_{jj'}^-$
Positive and negative deviations of crop j calculated from ECDF	δ_{jo}^+, δ_{jo}^-

Parameters:

Uncertain parameters of the SP problem (prices)	x_{jo}
Mean values of price dataset of crop j	$Mean_j$
Variance values of price dataset of crop j	Var_j
Co-variance values of price dataset between crop j and crop j'	$CoV_{jj'}$
Standard deviation value of crop j	Std_j
Number of observations of crop j dataset	n_j
Weight value of variance of crop j price	w_var_j
Weight value of co-variance crop j and crop j'	$w_{cov\,jj'}$
Weight value of deviations of crop j	w_{jo}

Objective

$$\min Z = \mu + \gamma + \xi$$

S.t.

$$\sum_{o=1}^{O} prob_o = 1$$

$$\sum_{o=1}^{O} (x_{jo} \times prob_o) = Mean_j$$

$$\sum_{o=1}^{O} (x_{jo} - Mean_j)^2 prob_o + var_j^+ - var_j^- = Var_j$$

$$\sum_{o=1}^{O} (x_{jo} - Mean_j)(x_{j'o} - Mean_{j'}) prob_o + cov_{jj'}^+ - cov_{jj'}^- = CoV_{jj'}$$

$$\widehat{ECDF}(x_{jo}) - \sum_{o'=1}^{o} prob_{o'} = \delta_{jo}^+ - \delta_{jo}^- \qquad o = 1 \dots O$$

With

$$\widehat{ECDF}(x_{jo}) = \Phi\left[\frac{x_{jo} - Mean_j}{\sqrt{Var_j}}\right]$$

$$\mu \geq w\_var_j \times v_j^+$$

$$\mu \geq w\_var_j \times v_j^-$$

$$\gamma \geq w\_cov_{jj'} \times cov^{+}_{jj'}$$

$$\gamma \geq w\_cov_{jj'} \times cov^{-}_{jj'}$$

$$\xi \geq \omega_{jo} \times \delta^{+}_{jo}$$

$$\xi \geq \omega_{jo} \times \delta^{-}_{jo}$$

$$var^{+}_{j}, var^{-}_{j}, cov^{+}_{jj'}, cov^{-}_{jj'}, \delta^{+}_{jo}, \delta^{-}_{jo} \geq 0$$

$$prob_{o} \in [0,1]$$

$$x_{jo} \leq x_{j,o+1}$$

APPENDIX C1: PYTHON CODES OF THE DETERMINISTIC MODEL

- dm.py

```
from __future__ import annotations

from dataclasses import (
    dataclass,
    field,
)

import pulp

import constants
from lp import (
    LpModel,
    LpParams,
)


@dataclass(frozen=True)
class DMParams(LpParams):
    """Parameter set for the SP model.

    Args:
        crop_count(int): Number of crops.
        age_count(int): Number of ages.
        trader_count(int): Number of traders.
        market_count(int): Number of markets.
        year_count(int): Number of years.
        area_total(int): Total areas of the farm.
        costs_water(float): Costs of water.
        costs_light(float): Costs of light.
        costs_harvest(list[float]): Costs of harvesting, indexed by year.
        costs_cut(list[float]): Costs of cutting, indexed by year.
        costs_process(list[float]): Costs of processing, indexed by year.
        costs_worker_fixed(list[float]): Costs of a fixed worker, indexed by year.
        costs_worker_hired(list[float]): Costs of a hired worker, indexed by year.
        amounts_water_required(list[float]): Amounts of water required, indexed by age.
            Defaults to None.
        amounts_light_required(list[float]): Amounts of light required, indexed by crop.
            Defaults to None.

        areas_initial(list[list[float]]): Initial areas, indexed by crop and age.
        costs_plant(list[list[float]]): Costs of planting, indexed by year and crop.
        amounts_demand_byproduct(list[list[float]]): Amounts of by-products on demand,
```

```
        indexed by year and crop.
    productivities(list[list[list[float]]]): Productivities of each crop, indexed by
        year, crop and age.
    penalties_demand_trader(list[list[list[float]]]): Penalties if trade demand is
        not met, indexed by year, crop and trader.
    penalties_demand_market(list[list[list[float]]]): Penalties if market demand is
        not met, indexed by year, crop and market.
    amounts_demand_trader(list[list[list[float]]]): Amounts for trader on demand,
        indexed by year, crop and trader.
    amounts_demand_market(list[list[list[float]]]): Amounts for market on demand,
        indexed by year, crop and market.
    prices_byproduct(list[list[list[float]]]): Prices of by-products, indexed by
        year, crop and scenario.
    prices_trader(list[list[list[list[float]]]]): Prices of traders, indexed by
        year, crop, trader and scenario.
    prices_market(list[list[list[list[float]]]]): Prices of markets, indexed by
        year, crop, market and scenario.
    count_worker_plant(list[int], optional): Number of workers to plant, indexed by
        year. Defaults to None.
    count_worker_harvest(list[int], optional): Number of workers to harvest,
        indexed by year. Defaults to None.
    count_worker_cut(list[int], optional): Number of workers to cut, indexed by
        year. Defaults to None.
    amounts_water_max(float, optional): Maximum amounts of water. Defaults to None.
    amounts_light_max(float, optional): Maximum amounts of light. Defaults to None.
    areas_plant_min(list[list[float]], optional): Minimum areas to plant, indexed
        by year and crop. Defaults to None.

Returns:
    DMParams: Parameter set for the Deterministic model.
"""

crop_count: int = field(init=False)
age_count: int = field(init=False)
trader_count: int = field(init=False)
market_count: int = field(init=False)
year_count: int = field(init=False)
area_total: int
costs_water: float
costs_light: float
costs_harvest: list[float]
costs_plant: list[list[float]]
costs_cut: list[float]
costs_process: list[float]
costs_worker_fixed: list[float]
costs_worker_hired: list[float]
```

```
    amounts_water_required: list[float]
    amounts_light_required: list[float]
    areas_initial: list[list[float]]
    amounts_demand_trader: list[list[list[float]]]
    amounts_demand_market: list[list[list[float]]]
    amounts_demand_byproduct: list[list[float]]
    productivities: list[list[list[float]]]
    penalties_demand_trader: list[list[list[float]]]
    penalties_demand_market: list[list[list[float]]]
    prices_byproduct: list[list[list[float]]]
    prices_trader: list[list[list[list[float]]]]
    prices_market: list[list[list[list[float]]]]
    count_worker_plant: list[int] = None
    count_worker_harvest: list[int] = None
    count_worker_cut: list[int] = None
    areas_plant_min: list[list[float]] = None

    def __post_init__(self) -> None:
        """Post initialization of the SpParams class."""
        object.__setattr__(self, "crop_count", len(self.areas_initial))
        object.__setattr__(self, "age_count", len(self.areas_initial[0]))
        object.__setattr__(self, "trader_count", len(self.prices_trader[0][0]))
        object.__setattr__(self, "market_count", len(self.prices_market[0][0]))
        object.__setattr__(self, "year_count", len(self.costs_harvest)-1)


class DModel(LpModel[DMParams]):
    """Deterministic model.

    This class is a concrete implementation of the LpBase class for the SP model. It
    takes a parameter set SpParams as input.
    """

    def __init__(self, params: DMParams) -> None:
        """Initialize the SpModel class.

        Args:
            params (DMParams): Parameter set for the Deterministic model.
        """
        super().__init__(params)

    def build_problem(self) -> pulp.LpProblem:
        """Build the Deterministic problem.

        Returns:
            pulp.LpProblem: DM problem.
```

```
"""
# Variables
amounts_trader_shipped = pulp.LpVariable.dict(
    "amounts_trader_shipped",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
        range(self.params.trader_count),
    ),
    lowBound=0,
)
amounts_trader_shipped_under = pulp.LpVariable.dict(
    "amounts_trader_shipped_under",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
        range(self.params.trader_count),
    ),
    lowBound=0,
)
amounts_market_shipped = pulp.LpVariable.dict(
    "amounts_market_shipped",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
        range(self.params.market_count),
),
    lowBound=0,
)
amounts_market_shipped_under = pulp.LpVariable.dict(
    "amounts_market_shipped_under",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
        range(self.params.market_count),
),
    lowBound=0,
)
amounts_byproduct = pulp.LpVariable.dict(
    "amounts_byproduct",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
    ),
    lowBound=0,
)
```

```
areas = pulp.LpVariable.dict(
    "areas",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
        range(self.params.age_count),
    ),
    lowBound=0,
)
areas_plant = pulp.LpVariable.dict(
    "areas_plant",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
    ),
    lowBound=0,
)
areas_cut = pulp.LpVariable.dict(
    "areas_cut",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
        range(self.params.age_count),
    ),
    lowBound=0,
)
count_worker_fixed = pulp.LpVariable.dict(
    "count_worker_fixed",
    range(self.params.year_count+1),
    lowBound=0,
    cat="Integer",
)
count_worker_hired = pulp.LpVariable.dict(
    "count_worker_hired",
    range(self.params.year_count+1),
    lowBound=0,
    cat="Integer",
)
self.vars.update(
    {
        "amounts_trader_shipped": amounts_trader_shipped,
        "amounts_trader_shipped_under": amounts_trader_shipped_under,
        "amounts_market_shipped": amounts_market_shipped,
        "amounts_market_shipped_under": amounts_market_shipped_under,
        "amounts_byproduct": amounts_byproduct,
        "areas": areas,
```

```
        "areas_plant": areas_plant,
        "areas_cut": areas_cut,
        "count_worker_fixed": count_worker_fixed,
        "count_worker_hired": count_worker_hired,
    }
)

# Objective
problem = pulp.LpProblem("problem", pulp.LpMaximize)
addend_1 = 0

for year in range(1, self.params.year_count + 1):
    addend_1 += (
        pulp.lpSum(
            self.params.prices_trader[year][crop][trader]
            * amounts_trader_shipped[year, crop, trader]
            for crop in range(self.params.crop_count)
            for trader in range(self.params.trader_count)
        )
        + pulp.lpSum(
            self.params.prices_market[year][crop][market]
            * amounts_market_shipped[year, crop, market]
            for crop in range(self.params.crop_count)
            for market in range(self.params.market_count)
        )
        + pulp.lpSum(
            self.params.prices_byproduct[year][crop]
            * amounts_byproduct[year, crop]
            for crop in range(self.params.crop_count)
        )
        - pulp.lpSum(
            self.params.penalties_demand_trader[year][crop][trader]
            * amounts_trader_shipped_under[year, crop, trader]
            for crop in range(self.params.crop_count)
            for trader in range(self.params.trader_count)
        )
        - pulp.lpSum(
            self.params.penalties_demand_market[year][crop][market]
            * amounts_market_shipped_under[year, crop, market]
            for crop in range(self.params.crop_count)
            for market in range(self.params.market_count)
        )
        - self.params.costs_process[year]
        * pulp.lpSum(
            amounts_byproduct[year, crop]
            for crop in range(self.params.crop_count)
```

```
            )
        )
    addend_2 = -pulp.lpSum(
        self.params.costs_plant[year][crop] * areas_plant[year, crop]
        for crop in range(self.params.crop_count)
        for age in range(self.params.age_count)
    )
    addend_3 = -self.params.costs_harvest[year] * pulp.lpSum(
        areas[year, crop, age]
        for crop in range(self.params.crop_count)
        for age in range(self.params.age_count)
    )
    addend_4 = -self.params.costs_cut[year] * pulp.lpSum(
        areas_cut[year, crop, age]
        for crop in range(self.params.crop_count)
        for age in range(self.params.age_count)
    )
    addend_5 = -self.params.costs_worker_fixed[year] * count_worker_fixed[year]
    addend_6 = -self.params.costs_worker_hired[year] * count_worker_hired[year]
    addend_7 = -self.params.costs_water * pulp.lpSum(
        self.params.amounts_water_required[age] * areas[year, crop, age]
        for crop in range(self.params.crop_count)
        for age in range(self.params.age_count)
    )
    addend_8 = -self.params.costs_light * pulp.lpSum(
        self.params.amounts_light_required[crop] * areas[year, crop, age]
        for crop in range(self.params.crop_count)
        for age in range(self.params.age_count)
    )
problem += (
    addend_1
    + addend_2
    + addend_3
    + addend_4
    + addend_5
    + addend_6
    + addend_7
    + addend_8,
    "objective",
)

# Constraints
for year in range(1, self.params.year_count+1):
    problem += (
        pulp.lpSum(
            areas[year, crop, age]
```

```
        for crop in range(self.params.crop_count)
        for age in range(self.params.age_count)
    )
    <= self.params.area_total,
    f"cons_areas_used_year_{year}",
)
for crop in range(self.params.crop_count):
    problem += (
        areas[year, crop, 0] == areas_plant[year, crop],
        f"cons_areas_crop_{crop}_age_0_year_{year}",
    )
for crop in range(self.params.crop_count):
    problem += (
        areas_cut[year, crop, 0] == 0,
        f"cons_areas_cut_crop_{crop}_age_first_year_{year}",
    )
if year == 1:
    for crop in range(self.params.crop_count):
        for age in range(1, self.params.age_count):
            problem += (
                areas[year, crop, age]
                == self.params.areas_initial[crop][age - 1]
                - areas_cut[year, crop, age],
                f"cons_areas_crop_{crop}_age_{age}_year_{year}",
            )
    for crop in range(self.params.crop_count):
        problem += (
            areas_cut[year, crop, self.params.age_count - 1]
            == self.params.areas_initial[crop][
                self.params.age_count - 1 - 1
            ],
            f"cons_areas_cut_crop_{crop}_age_last_year_{year}",
        )
else:
    for crop in range(self.params.crop_count):
        for age in range(1, self.params.age_count):
            problem += (
                areas[year, crop, age]
                == areas[year - 1, crop, age - 1]
                - areas_cut[year, crop, age],
                f"cons_areas_crop_{crop}_age_{age}_year_{year}",
            )
    for crop in range(self.params.crop_count):
        problem += (
            areas_cut[year, crop, self.params.age_count - 1]
            == areas[year - 1, crop, self.params.age_count - 1 - 1],
```

```
            f"cons_areas_cut_crop_{crop}_age_last_year_{year}",
        )
for crop in range(self.params.crop_count):
    problem += (
        pulp.lpSum(
            amounts_trader_shipped[year, crop, trader]
            for trader in range(self.params.trader_count)
        )
        + pulp.lpSum(
            amounts_market_shipped[year, crop, market]
            for market in range(self.params.market_count)
        )
        + amounts_byproduct[year, crop]
        <= pulp.lpSum(
            self.params.productivities[year][crop][age]
            * areas[year, crop, age]
            for age in range(self.params.age_count)
        ),
        "cons_productivities_"
        f"crop_{crop}_year_{year}",
    )
for crop in range(self.params.crop_count):
    for trader in range(self.params.trader_count):
        problem += (
            amounts_trader_shipped[year, crop, trader]
            == self.params.amounts_demand_trader[year][crop][trader]
            - amounts_trader_shipped_under[
                year, crop, trader
            ],
            "cons_amounts_"
            f"crop_{crop}_"
            f"trader_{trader}_"
            f"year_{year}",
        )
for crop in range(self.params.crop_count):
    for market in range(self.params.market_count):
        problem += (
            amounts_market_shipped[year, crop, market]
            == self.params.amounts_demand_market[year][crop][market]
            - amounts_market_shipped_under[
                year, crop, market
            ],
            "cons_amounts_"
            f"crop_{crop}_"
            f"market_{market}_"
            f"year_{year}",
```

```
        )
    for crop in range(self.params.crop_count):
        problem += (
            amounts_byproduct[year, crop]
            <= self.params.amounts_demand_byproduct[year][crop],
            "cons_amounts_byproduct_"
            f"crop_{crop}_year_{year}",
        )

    problem += (
        count_worker_fixed[year]
        <= 5,
        "cons_count_worker_fixed_"
        f"year_{year}",
    )
    problem += (
        count_worker_hired [year]
        <= 1000,
        "cons_count_worker_hired_"
        f"year_{year}",
    )
    total_worker_area_plant = pulp.lpSum(
        self.params.count_worker_plant[year] *
        areas_plant[year, crop]
        for crop in range(self.params.crop_count)
    )
    total_worker_area_harvest = pulp.lpSum(
        self.params.count_worker_harvest[year] *
        areas[year, crop, age]
        for crop in range(self.params.crop_count)
        for age in range(self.params.age_count)
    )
    total_worker_area_cut = pulp.lpSum(
        self.params.count_worker_cut[year] *
        areas_cut[year, crop, age]
        for crop in range(self.params.crop_count)
        for age in range(self.params.age_count)
    )
    problem += (
        count_worker_fixed[year] + count_worker_hired[year] ==
        total_worker_area_plant +
        total_worker_area_harvest +
        total_worker_area_cut,
        "cons_labour_requirement_"
        f"year_{year}",
    )
```

```
    return problem

def calculate_results(self) -> dict:
        """Calculate yield and average metrics after solving the model."""
        # Retrieve the total profit from the solved objective value
        total_profit = self.problem.objective.value()

        # Calculate total cost including penalties and total yield
        total_cost = 0
        total_yield = 0

        for year in range(1, self.params.year_count + 1):
            # Accumulate total cost including penalties
            total_cost += (
                # addend_1
                sum(
                self.params.penalties_demand_trader[year][crop][trader]
                * self.vars["amounts_trader_shipped_under"][year, crop, trader].varValue
                for crop in range(self.params.crop_count)
                for trader in range(self.params.trader_count)
                )
                    + sum(
                self.params.penalties_demand_market[year][crop][market]
                * self.vars["amounts_market_shipped_under"][year, crop, market].varValue
                for crop in range(self.params.crop_count)
                for market in range(self.params.market_count)
                )
                    + self.params.costs_process[year] * sum(
                self.vars["amounts_byproduct"][year, crop].varValue
                for crop in range(self.params.crop_count)
                )
                # addend_2
                + sum(self.params.costs_plant[year][crop] *
                self.vars["areas_plant"][year, crop].varValue
                for crop in range(self.params.crop_count)
                )
                # addend_3
                    + self.params.costs_harvest[year] * sum(
                self.vars["areas"][year, crop, age].varValue
                for crop in range(self.params.crop_count)
                for age in range(self.params.age_count)
                )
                # addend_4
                    + self.params.costs_cut[year] * sum(
                self.vars["areas_cut"][year, crop, age].varValue
                for crop in range(self.params.crop_count)
```

```
                for age in range(self.params.age_count)
                )
                # addend_5
                  +                    self.params.costs_worker_fixed[year]                    *
self.vars["count_worker_fixed"][year].varValue
                # addend_6
                  +                    self.params.costs_worker_hired[year]                    *
self.vars["count_worker_hired"][year].varValue
                # addend_7
                  + self.params.costs_water * sum(
                self.params.amounts_water_required[age]   *   self.vars["areas"][year,   crop,
age].varValue
                for crop in range(self.params.crop_count)
                for age in range(self.params.age_count)
                )
                # addend_8
                  + self.params.costs_light * sum(
                self.params.amounts_light_required[crop]   *   self.vars["areas"][year,   crop,
age].varValue
                for crop in range(self.params.crop_count)
                for age in range(self.params.age_count)
                )
            )
            total_yield += sum(
                self.params.productivities[year][crop][age]   *   self.vars["areas"][year,   crop,
age].varValue
                for crop in range(self.params.crop_count)
                for age in range(self.params.age_count)
            )

        average_yield_per_hectare_per_year = (
            total_yield / (self.params.area_total * self.params.year_count) if total_yield > 0
else 0
        )
        average_cost_per_hectare_per_year    =    total_cost    /    (self.params.area_total    *
self.params.year_count)
        average_cost_per_kg_per_year = total_cost / (total_yield * 1000) if total_yield > 0
else 0
        average_profit_per_hectare_per_year   =   total_profit   /   (self.params.area_total   *
self.params.year_count)
        average_profit_per_kg_per_year = total_profit / (total_yield * 1000) if total_yield
> 0 else 0

        # Return results in a dictionary
        print("Total profit:", total_profit)
        print("Total cost:", total_cost)
```

```
        print("Total yield (tons):", total_yield)
        print("Average yield per hectare per year (tons/ha/year):", average_yield_per_hectare_per_year)
        print("Average cost per hectare per year:", average_cost_per_hectare_per_year)
        print("Average cost per kilogram per year:", average_cost_per_kg_per_year)
        print("Average profit per hectare per year:", average_profit_per_hectare_per_year)
        print("Average profit per kilogram per year:", average_profit_per_kg_per_year)
```

- problems.py

```
from constants import *
import pandas as pd
from scipy.stats import norm
import random
import time
from dm import (
    DModel,
    DMParams,
)

print('Deterministic model')
print(f"Number of years: {year_count}")

records = []

finalResults = []
penalties_demand_trader = []
penalties_demand_market = []

for year in range(year_count + 1):
    year_data_penalties_demand_trader = []
    year_data_penalties_demand_market = []

    for crop in range(crop_count):
        prices = prices_trader[year][crop]
        crop_data_penalties_demand_trader = [p*0.1 for p in prices]
        crop_data_penalties_demand_market = [p*0.1 for p in prices]

        year_data_penalties_demand_trader.append(crop_data_penalties_demand_trader)
        year_data_penalties_demand_market.append(crop_data_penalties_demand_market)

    penalties_demand_trader.append(year_data_penalties_demand_trader)
    penalties_demand_market.append(year_data_penalties_demand_market)

# Compute expected profit using SPModel
dm_params = DMParams(
```

```
    area_total=area_total,
    costs_water=costs_water,
    costs_light=costs_light,
    costs_harvest=costs_harvest,
    costs_cut=costs_cut,
    costs_process=costs_process,
    costs_worker_fixed=costs_worker_fixed,
    costs_worker_hired=costs_worker_hired,
    amounts_water_required=amounts_water_required,
    amounts_light_required=amounts_light_required,

    areas_initial=areas_initial,
    costs_plant=costs_plant,
    amounts_demand_byproduct=amounts_demand_byproduct,
    productivities=productivities,
    penalties_demand_trader=penalties_demand_trader,
    penalties_demand_market=penalties_demand_market,
    amounts_demand_trader=amounts_demand_trader,
    amounts_demand_market=amounts_demand_market,
    prices_byproduct=prices_byproduct,
    prices_trader=prices_trader,
    prices_market=prices_market,
    count_worker_plant=count_worker_plant,
    count_worker_harvest=count_worker_harvest,
    count_worker_cut=count_worker_cut
)
dm_model = DModel(dm_params)
# print("Begin to solve")
# Record the start time
start_time = time.time()
dm_model.solve()
# Record the end time
end_time = time.time()

# Calculate the elapsed time
elapsed_time = end_time - start_time

# Print the elapsed time
print(f"Elapsed Time: {elapsed_time} seconds")

c = dm_model.problem.constraints
print(f"Constraints {len(c)}")
v = dm_model.problem.variables()
print(f"Variables {len(v)}")

profit = dm_model.problem.objective.value()
```

```
# print(f"Profit: {profit}")
finalResults.append(profit)

# print("Profits")
# print(",".join([str(e) for e in finalResults]))


# Calculate and display results
results = dm_model.calculate_results()
print(results)
```

- constants.py

```
year_count = 8
crop_count = 3
trader_count = 5
market_count = 5

area_total = 20
costs_water = 0.1
costs_light = 0.15
costs_harvest = [1700] * (year_count+1)
costs_plant = [[5000] * crop_count] * (year_count+1)
costs_cut = [3000] * (year_count+1)
costs_process = [300] * (year_count+1)
costs_worker_fixed = [5000] * (year_count+1)
costs_worker_hired = [1700] * (year_count+1)
amounts_water_required = [755, 1235, 1925, 2585, 2585, 2585, 2585, 2585, 2585, 2585]
amounts_light_required = [750] * crop_count
areas_initial = [
    [0, 0, 0, 0, 20, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 20, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
]
productivities = [
    [
        [0, 50, 60, 80, 90, 90, 90, 80, 50, 30],
        [0, 40, 50, 70, 80, 80, 80, 70, 40, 20],
        [0, 10, 20, 30, 30, 30, 30, 20, 10, 10],
    ]
] * (year_count+1)
amounts_demand_byproduct = [[20, 0, 0]] * (year_count+1)

amounts_demand_trader = [
    [[200] * trader_count, [160] * trader_count, [40] * trader_count]
] * (year_count+1)
```

```
amounts_demand_market = [
    [[200] * market_count, [160] * market_count, [40] * market_count]
] * (year_count+1)

prices_trader = [
  [[900] * trader_count, [1500] * trader_count, [3000] * trader_count]] * (year_count+1)

prices_market = [
    [[900] * market_count, [1500] * market_count, [3000] * market_count]
 ] * (year_count+1)

prices_byproduct = [[300, 0, 0]] * (year_count+1)

count_worker_plant = [10,0,0,0,0,0,0,0,0,0]

count_worker_harvest = [0,3,3,3,3,3,3,3,3,0]

count_worker_cut = [0,0,0,0,0,0,0,0,0,6]
```

APPENDIX C2: PYTHON CODES OF SCENARIO TREE GENERATION METHOD

- sp.py

```
from __future__ import annotations
from app.solution.data_structures import Storage

from dataclasses import (
    dataclass,
    field,
)

import pulp

from spfp.lp import (
    LpModel,
    LpParams,
)


@dataclass(frozen=True)
class SpParams(LpParams):
    """Parameter set for the SP model.

    Args:
        crop_count(int): Number of crops.
        age_count(int): Number of ages.
        trader_count(int): Number of traders.
        market_count(int): Number of markets.
        scenario_count(int): Number of scenarios.
        year_count(int): Number of years.
        area_total(int): Total areas of the farm.
        costs_water(float): Costs of water.
        costs_light(float): Costs of light.
        costs_harvest(list[float]): Costs of harvesting, indexed by year.
        costs_cut(list[float]): Costs of cutting, indexed by year.
        costs_process(list[float]): Costs of processing, indexed by year.
        costs_worker_fixed(list[float]): Costs of a fixed worker, indexed by year.
        costs_worker_hired(list[float]): Costs of a hired worker, indexed by year.
        amounts_water_required(list[float]): Amounts of water required, indexed by age.
            Defaults to None.
        amounts_light_required(list[float]): Amounts of light required, indexed by crop.
            Defaults to None.
        probabilities(list[list[float]]): Probabilities of each scenario, indexed by
```

```
        year and scenario.
    areas_initial(list[list[float]]): Initial areas, indexed by crop and age.
    costs_plant(list[list[float]]): Costs of planting, indexed by year and crop.
    amounts_demand_byproduct(list[list[float]]): Amounts of by-products on demand,
        indexed by year and crop.
    productivities(list[list[list[float]]]): Productivities of each crop, indexed by
        year, crop and age.
    penalties_demand_trader(list[list[list[float]]]): Penalties if trade demand is
        not met, indexed by year, crop and trader.
    penalties_demand_market(list[list[list[float]]]): Penalties if market demand is
        not met, indexed by year, crop and market.
    amounts_demand_trader(list[list[list[float]]]): Amounts for trader on demand,
        indexed by year, crop and trader.
    amounts_demand_market(list[list[list[float]]]): Amounts for market on demand,
        indexed by year, crop and market.
    prices_byproduct(list[list[list[float]]]): Prices of by-products, indexed by
        year, crop and scenario.
    prices_trader(list[list[list[list[float]]]]): Prices of traders, indexed by
        year, crop, trader and scenario.
    prices_market(list[list[list[list[float]]]]): Prices of markets, indexed by
        year, crop, market and scenario.
    count_worker_plant(list[int], optional): Number of workers to plant, indexed by
        year. Defaults to None.
    count_worker_harvest(list[int], optional): Number of workers to harvest,
        indexed by year. Defaults to None.
    count_worker_cut(list[int], optional): Number of workers to cut, indexed by
        year. Defaults to None.
    amounts_water_max(float, optional): Maximum amounts of water. Defaults to None.
    amounts_light_max(float, optional): Maximum amounts of light. Defaults to None.
    areas_plant_min(list[list[float]], optional): Minimum areas to plant, indexed
        by year and crop. Defaults to None.

Returns:
    SpParams: Parameter set for the SP model.
"""

crop_count: int = field(init=False)
age_count: int = field(init=False)
trader_count: int = field(init=False)
market_count: int = field(init=False)
scenario_count: int = field(init=False)
year_count: int = field(init=False)
area_total: int
costs_water: float
costs_light: float
costs_harvest: list[float]
```

```
    costs_cut: list[float]
    costs_process: list[float]
    costs_worker_fixed: list[float]
    costs_worker_hired: list[float]
    amounts_water_required: list[float]
    amounts_light_required: list[float]
    probabilities: list[list[float]]
    areas_initial: list[list[float]]
    costs_plant: list[list[float]]
    amounts_demand_byproduct: list[list[float]]
    productivities: list[list[list[float]]]
    penalties_demand_trader: list[list[list[float]]]
    penalties_demand_market: list[list[list[float]]]
    amounts_demand_trader: list[list[list[float]]]
    amounts_demand_market: list[list[list[float]]]
    prices_byproduct: list[list[list[float]]]
    prices_trader: list[list[list[list[float]]]]
    prices_market: list[list[list[list[float]]]]
    storage: Storage
    count_worker_plant: list[float]
    count_worker_harvest: list[float]
    count_worker_cut: list[float]
    areas_plant_min: list[list[float]] = None

    def __post_init__(self) -> None:
        """Post initialization of the SpParams class."""
        object.__setattr__(self, "crop_count", len(self.areas_initial))
        object.__setattr__(self, "age_count", len(self.areas_initial[0]))
        object.__setattr__(self, "trader_count", len(self.prices_trader[0][0]))
        object.__setattr__(self, "market_count", len(self.prices_market[0][0]))
        object.__setattr__(self, "scenario_count", len(self.probabilities[-1]))
        object.__setattr__(self, "year_count", len(self.costs_harvest)-1)

        # Initialize worker counts if they are None
        if self.count_worker_plant is None:
            object.__setattr__(self, "count_worker_plant", [0] * (self.year_count + 1))
        if self.count_worker_harvest is None:
            object.__setattr__(self, "count_worker_harvest", [0] * (self.year_count + 1))
        if self.count_worker_cut is None:
            object.__setattr__(self, "count_worker_cut", [0] * (self.year_count + 1))


class SpModel(LpModel[SpParams]):
    """SP model.

    This class is a concrete implementation of the LpBase class for the SP model. It
```

```
    takes a parameter set SpParams as input.
    """

    def __init__(self, params: SpParams) -> None:
        """Initialize the SpModel class.

        Args:
            params (SpParams): Parameter set for the SP model.
        """
        super().__init__(params)

    def build_problem(self) -> pulp.LpProblem:
        """Build the SP problem.

        Returns:
            pulp.LpProblem: SP problem.
        """
        # Variables
        amounts_trader_shipped = pulp.LpVariable.dict(
            "amounts_trader_shipped",
            (
                range(self.params.year_count+1),
                range(self.params.crop_count),
                range(self.params.trader_count),
                range(self.params.scenario_count),
            ),
            lowBound=0,
        )
        amounts_trader_shipped_under = pulp.LpVariable.dict(
            "amounts_trader_shipped_under",
            (
                range(self.params.year_count+1),
                range(self.params.crop_count),
                range(self.params.trader_count),
                range(self.params.scenario_count),
            ),
            lowBound=0,
        )
        amounts_market_shipped = pulp.LpVariable.dict(
            "amounts_market_shipped",
            (
                range(self.params.year_count+1),
                range(self.params.crop_count),
                range(self.params.market_count),
                range(self.params.scenario_count),
            ),
```

```
    lowBound=0,
)
amounts_market_shipped_under = pulp.LpVariable.dict(
    "amounts_market_shipped_under",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
        range(self.params.market_count),
        range(self.params.scenario_count),
    ),
    lowBound=0,
)
amounts_byproduct = pulp.LpVariable.dict(
    "amounts_byproduct",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
        range(self.params.scenario_count),
    ),
    lowBound=0,
)
areas = pulp.LpVariable.dict(
    "areas",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
        range(self.params.age_count),
    ),
    lowBound=0,
)
areas_plant = pulp.LpVariable.dict(
    "areas_plant",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
    ),
    lowBound=0,
)
areas_cut = pulp.LpVariable.dict(
    "areas_cut",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
        range(self.params.age_count),
    ),
    lowBound=0,
```

```
)
count_worker_fixed = pulp.LpVariable.dict(
    "count_worker_fixed",
    range(self.params.year_count+1),
    lowBound=0,
    cat="Integer",
)
count_worker_hired = pulp.LpVariable.dict(
    "count_worker_hired",
    range(self.params.year_count+1),
    lowBound=0,
    cat="Integer",
)
self.vars.update(
    {
        "amounts_trader_shipped": amounts_trader_shipped,
        "amounts_trader_shipped_under": amounts_trader_shipped_under,
        "amounts_market_shipped": amounts_market_shipped,
        "amounts_market_shipped_under": amounts_market_shipped_under,
        "amounts_byproduct": amounts_byproduct,
        "areas": areas,
        "areas_plant": areas_plant,
        "areas_cut": areas_cut,
        "count_worker_fixed": count_worker_fixed,
        "count_worker_hired": count_worker_hired,
    }
)

# Objective
problem = pulp.LpProblem("problem", pulp.LpMaximize)
addend_1 = 0

# What is amounts_trader_shipped
for year in range(1, self.params.year_count + 1):
    for scenario_index in range(len(self.params.storage.years[year])):
        addend_1 += self.params.probabilities[year][scenario_index] * (
            pulp.lpSum(
                self.params.prices_trader[year][crop][trader][scenario_index]
                * amounts_trader_shipped[year, crop, trader, scenario_index]
                for crop in range(self.params.crop_count)
                for trader in range(self.params.trader_count)
            )
            + pulp.lpSum(
                self.params.prices_market[year][crop][market][scenario_index]
                * amounts_market_shipped[year, crop, market, scenario_index]
                for crop in range(self.params.crop_count)
```

```
                    for market in range(self.params.market_count)
                )
                + pulp.lpSum(
                    self.params.prices_byproduct[year][crop][scenario_index]
                    * amounts_byproduct[year, crop, scenario_index]
                    for crop in range(self.params.crop_count)
                )
                - pulp.lpSum(
                    self.params.penalties_demand_trader[year][crop][trader][scenario_index]
                    * amounts_trader_shipped_under[year, crop, trader, scenario_index]
                    for crop in range(self.params.crop_count)
                    for trader in range(self.params.trader_count)
                )
                - pulp.lpSum(

self.params.penalties_demand_market[year][crop][market][scenario_index]
                    * amounts_market_shipped_under[year, crop, market, scenario_index]
                    for crop in range(self.params.crop_count)
                    for market in range(self.params.market_count)
                )
                - self.params.costs_process[year]
                * pulp.lpSum(
                    amounts_byproduct[year, crop, scenario_index]
                    for crop in range(self.params.crop_count)
                )
            )
        addend_2 = -pulp.lpSum(
            self.params.costs_plant[year][crop] * areas_plant[year, crop]
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )
        addend_3 = -self.params.costs_harvest[year] * pulp.lpSum(
            areas[year, crop, age]
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )
        addend_4 = -self.params.costs_cut[year] * pulp.lpSum(
            areas_cut[year, crop, age]
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )
        addend_5 = -self.params.costs_worker_fixed[year] * count_worker_fixed[year]
        addend_6 = -self.params.costs_worker_hired[year] * count_worker_hired[year]
        addend_7 = -self.params.costs_water * pulp.lpSum(
            self.params.amounts_water_required[age] * areas[year, crop, age]
            for crop in range(self.params.crop_count)
```

```
        for age in range(self.params.age_count)
    )
    addend_8 = -self.params.costs_light * pulp.lpSum(
        self.params.amounts_light_required[crop] * areas[year, crop, age]
        for crop in range(self.params.crop_count)
        for age in range(self.params.age_count)
    )
problem += (
    addend_1
    + addend_2
    + addend_3
    + addend_4
    + addend_5
    + addend_6
    + addend_7
    + addend_8,
    "objective",
)

# Constraints
for year in range(1, self.params.year_count+1):
    problem += (
        pulp.lpSum(
            areas[year, crop, age]
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )
        <= self.params.area_total,
        f"cons_areas_used_year_{year}",
    )
    for crop in range(self.params.crop_count):
        problem += (
            areas[year, crop, 0] == areas_plant[year, crop],
            f"cons_areas_crop_{crop}_age_0_year_{year}",
        )
    for crop in range(self.params.crop_count):
        problem += (
            areas_cut[year, crop, 0] == 0,
            f"cons_areas_cut_crop_{crop}_age_first_year_{year}",
        )
    if year == 1:
        for crop in range(self.params.crop_count):
            for age in range(1, self.params.age_count):
                problem += (
                    areas[year, crop, age]
                    == self.params.areas_initial[crop][age - 1]
```

```
            - areas_cut[year, crop, age],
            f"cons_areas_crop_{crop}_age_{age}_year_{year}",
        )
    for crop in range(self.params.crop_count):
        problem += (
            areas_cut[year, crop, self.params.age_count - 1]
            == self.params.areas_initial[crop][
                self.params.age_count - 1 - 1
            ],
            f"cons_areas_cut_crop_{crop}_age_last_year_{year}",
        )
else:
    for crop in range(self.params.crop_count):
        for age in range(1, self.params.age_count):
            problem += (
                areas[year, crop, age]
                == areas[year - 1, crop, age - 1]
                - areas_cut[year, crop, age],
                f"cons_areas_crop_{crop}_age_{age}_year_{year}",
            )
    for crop in range(self.params.crop_count):
        problem += (
            areas_cut[year, crop, self.params.age_count - 1]
            == areas[year - 1, crop, self.params.age_count - 1 - 1],
            f"cons_areas_cut_crop_{crop}_age_last_year_{year}",
        )
for crop in range(self.params.crop_count):
    for scenario in range(self.params.scenario_count):
        problem += (
            pulp.lpSum(
                amounts_trader_shipped[year, crop, trader, scenario]
                for trader in range(self.params.trader_count)
            )
            + pulp.lpSum(
                amounts_market_shipped[year, crop, market, scenario]
                for market in range(self.params.market_count)
            )
            + amounts_byproduct[year, crop, scenario]
            <= pulp.lpSum(
                self.params.productivities[year][crop][age]
                * areas[year, crop, age]
                for age in range(self.params.age_count)
            ),
            "cons_productivities_"
            f"crop_{crop}_scenario_{scenario}_year_{year}",
        )
```

```
for crop in range(self.params.crop_count):
    for trader in range(self.params.trader_count):
        for scenario in range(self.params.scenario_count):
            problem += (
                amounts_trader_shipped[year, crop, trader, scenario]
                == self.params.amounts_demand_trader[year][crop][trader]
                - amounts_trader_shipped_under[
                    year, crop, trader, scenario
                ],
                "cons_amounts_"
                f"crop_{crop}_"
                f"trader_{trader}_"
                f"scenario_{scenario}_"
                f"year_{year}",
            )
for crop in range(self.params.crop_count):
    for market in range(self.params.market_count):
        for scenario in range(self.params.scenario_count):
            problem += (
                amounts_market_shipped[year, crop, market, scenario]
                == self.params.amounts_demand_market[year][crop][market]
                - amounts_market_shipped_under[
                    year, crop, market, scenario
                ],
                "cons_amounts_"
                f"crop_{crop}_"
                f"market_{market}_"
                f"scenario_{scenario}_"
                f"year_{year}",
            )
for crop in range(self.params.crop_count):
    for scenario in range(self.params.scenario_count):
        problem += (
            amounts_byproduct[year, crop, scenario]
            <= self.params.amounts_demand_byproduct[year][crop],
            "cons_amounts_byproduct_"
            f"crop_{crop}_scenario_{scenario}_year_{year}",
        )
problem += (
    count_worker_fixed[year]
    <= 5,
    "cons_count_worker_fixed_"
    f"year_{year}",
)
problem += (
    count_worker_hired[year]
```

```
            <= 1000,
            "cons_count_worker_hired_"
            f"year_{year}",
        )
        total_worker_area_plant = pulp.lpSum(
            self.params.count_worker_plant[year] *
            areas_plant[year, crop]
            for crop in range(self.params.crop_count)
        )
        total_worker_area_harvest = pulp.lpSum(
            self.params.count_worker_harvest[year] *
            areas[year, crop, age]
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )
        total_worker_area_cut = pulp.lpSum(
            self.params.count_worker_cut[year] *
            areas_cut[year, crop, age]
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )
        problem += (
            count_worker_fixed[year] + count_worker_hired[year] ==
            total_worker_area_plant +
            total_worker_area_harvest +
            total_worker_area_cut,
            "cons_labour_requirement_"
            f"year_{year}",
        )
    return problem

def calculate_results(self):
    # Retrieve the total profit from the solved objective value
    total_profit = self.problem.objective.value()

    # Calculate total cost including penalties and total yield
    total_cost = 0
    total_yield = 0

    for year in range(1, self.params.year_count + 1):
        # Accumulate total cost including penalties
        total_cost += (
            # addend_1
            sum(
                self.params.penalties_demand_trader[year][crop][trader][scenario_index] *
```

```
            self.vars["amounts_trader_shipped_under"][year,          crop,          trader,
scenario_index].varValue
            for crop in range(self.params.crop_count)
            for trader in range(self.params.trader_count)
            for scenario_index in range(len(self.params.storage.years[year]))
        )
        +
sum(self.params.penalties_demand_market[year][crop][market][scenario_index] *
            self.vars["amounts_market_shipped_under"][year,        crop,        market,
scenario_index].varValue
            for crop in range(self.params.crop_count)
            for market in range(self.params.market_count)
            for scenario_index in range(len(self.params.storage.years[year]))
        )
        + self.params.costs_process[year] * sum(
            self.vars["amounts_byproduct"][year, crop,scenario_index].varValue
            for crop in range(self.params.crop_count)
            for scenario_index in range(len(self.params.storage.years[year]))
        )
        # addend_2
        +   sum(self.params.costs_plant[year][crop]   *   self.vars["areas_plant"][year,
crop].varValue
             for crop in range(self.params.crop_count)
        )
        # addend_3
        + self.params.costs_harvest[year] * sum(
        self.vars["areas"][year, crop, age].varValue
        for crop in range(self.params.crop_count)
        for age in range(self.params.age_count)
        )
        # addend_4
        + self.params.costs_cut[year] * sum(
        self.vars["areas_cut"][year, crop, age].varValue
        for crop in range(self.params.crop_count)
        for age in range(self.params.age_count)
        )
        # addend_5
        +                    self.params.costs_worker_fixed[year]                    *
self.vars["count_worker_fixed"][year].varValue

        # addend_6
        +                    self.params.costs_worker_hired[year]                    *
self.vars["count_worker_hired"][year].varValue

        # addend_7
        + self.params.costs_water * sum(
```

```
            self.params.amounts_water_required[age] * self.vars["areas"][year, crop, age].varValue
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
            )
            # addend_8
            + self.params.costs_light * sum(
            self.params.amounts_light_required[crop] * self.vars["areas"][year, crop, age].varValue
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
            )
        )

        total_yield += sum(
            self.params.productivities[year][crop][age] * self.vars["areas"][year, crop, age].varValue
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )

    # Calculating the derived metrics
    average_yield_per_hectare_per_year = total_yield / (
            self.params.area_total * self.params.year_count) if total_yield > 0 else 0
    average_cost_per_hectare_per_year = total_cost / (
            self.params.area_total * self.params.year_count) if total_yield > 0 else 0
    average_cost_per_kg_per_year = total_cost / (total_yield * 1000) if total_yield > 0 else 0
    average_profit_per_hectare_per_year = total_profit / (
            self.params.area_total * self.params.year_count) if total_yield > 0 else 0
    average_profit_per_kg_per_year = total_profit / (total_yield * 1000) if total_yield > 0 else 0

    # Return results in a dictionary
    print("Total profit:", total_profit)
    print("Total cost:", total_cost)
    print("Total yield (tons):", total_yield)
    print("Average yield per hectare per year (tons/ha/year):", average_yield_per_hectare_per_year)
    print("Average cost per hectare per year:", average_cost_per_hectare_per_year)
    print("Average cost per kilogram per year:", average_cost_per_kg_per_year)
    print("Average profit per hectare per year:", average_profit_per_hectare_per_year)
    print("Average profit per kilogram per year:", average_profit_per_kg_per_year)
```

- lp.py

```
"""Linear programing abstractions."""
```

```python
from abc import (
    ABC,
    abstractmethod,
)
from typing import (
    Generic,
    TypeVar,
)

import pulp


class LpParams(ABC):
    """Linear programing parameters."""


LpParamsT = TypeVar("LpParamsT", bound=LpParams)


class LpModel(ABC, Generic[LpParamsT]):
    """Abstract class for linear programming."""

    def __init__(self, params: LpParamsT) -> None:
        """Initialize the linear programming."""
        self.params = params
        self.vars: dict[str, dict[tuple[int], pulp.LpVariable]] = {}
        self.problem: pulp.LpProblem = None
        self.solver: pulp.apis.core.LpSolver = None

    @abstractmethod
    def build_problem(self) -> pulp.LpProblem:
        """Build the linear programming problem.

        Returns:
            pulp.LpProblem: The linear programming problem.
        """
        pass

    def build_solver(self) -> pulp.apis.core.LpSolver:
        """Build the linear programming solver.

        Returns:
            pulp.apis.core.LpSolver: The linear programming solver.
        """
        return pulp.PULP_CBC_CMD(msg=False)
```

```
    def solve(self) -> None:
        """Solve the linear programming."""
        if self.problem is None:
            self.problem = self.build_problem()
        if self.solver is None:
            self.solver = self.build_solver()
        self.problem.solve(self.solver)

    def get_vars_dict(self, name: str) -> dict[tuple[int], pulp.LpVariable]:
        """Get the variables dictionary with the given name.

        The key is a list of indexes and the value is the variable.

        Args:
            name (str): The name of the variables.

        Returns:
            dict[tuple[int], pulp.LpVariable]: The variables.
        """
        if name not in self.vars:
            return {}
        return self.vars[name]

    @property
    def status(self) -> str:
        """Return the status of the linear programming.

        Returns:
            str: The status of the linear programming.

        | LpStatus key       | string value | numerical value |
        | ------------------ | ------------ | --------------- |
        | LpStatusOptimal    | “Optimal”    | 1               |
        | LpStatusNotSolved  | “Not Solved” | 0               |
        | LpStatusInfeasible | “Infeasible” | -1              |
        | LpStatusUnbounded  | “Unbounded”  | -2              |
        | LpStatusUndefined  | “Undefined”  | -3              |
        """
        return pulp.LpStatus[self.problem.status]
```

- lpdmp

```
from __future__ import annotations

from dataclasses import (
    dataclass,
```

```
    field,
)

import pandas as pd
import pulp

from spfp.lp import (
    LpModel,
    LpParams,
)


@dataclass(frozen=True)
class LpdmpParams(LpParams):
    """Parameter set for the LPDMP model.

    This class is used to store the parameters for the LPDMP model. It provide a class
    method to create a LpdmpParams object from a dataframe of prices history.

    Args:
        weights (list[list[float]]): Weights matrix indexed by crop and moment.
        prices (list[list[float]]): Prices matrix indexed by crop and scenario.
        moments (list[list[float]]): Moments matrix indexed by crop and moment.
        covariances (dict[(int, int), float]): Covariances dictionary indexed by
            combinations of crops. Each pair of crops (a tuple containing two crop
            indexes with the lower index preceded) is associated with their covariances.
            Example: {(0, 1): -0.01, (0, 2): -0.01, (1, 2): -0.01}
        omegas (list[list[float]]): Omegas matrix indexed by crop and scenario.

    Returns:
        LpdmpParams: Parameter set for the LPDMP model.
    """

    scenario_count: int = field(init=False)
    moment_count: int = field(init=False)
    crop_count: int = field(init=False)
    weights: list[list[float]]
    prices: list[list[float]]
    moments: list[list[float]]
    covariances: dict[tuple[int, int], float]
    omegas: list[list[float]]

    def __post_init__(self) -> None:
        """Post initialization of the LpdmpParams class."""
        object.__setattr__(self, "scenario_count", len(self.prices[0]))
        object.__setattr__(self, "crop_count", len(self.moments))
```

```
        object.__setattr__(self, "moment_count", len(self.moments[0]))

    @classmethod
    def from_crop_prices_history(
        cls,
        df_crop_prices: pd.DataFrame,
        stdev_scale_factors: list[float] = [-1.5, 0.0, 1.5],
    ) -> LpdmpParams:
        """Create LpdmpParams from crop prices history dataframe.

        Args:
            df_crop_prices (pd.DataFrame): Crop prices history dataframe. The dataframe
                must multiple columns, each column representing a crop, and each row
                representing a set of prices for crops. The last row must be the current
                prices, this row will be used to calculate the prices parameters.
            stdev_scale_factors (list[float]): List of scale factors for the standard
                deviation for each scenario. The scale factors will be applied to the
                current prices to create the prices in other scenarios. Default is
                [-1.5, 0.0, 1.5].
        Returns:
            LpdmpParams: Parameter set for the LPDMP model.
        """
        from itertools import combinations

        moment_count = 4
        scenario_count = len(stdev_scale_factors)
        crop_count = df_crop_prices.shape[1]
        current_prices = df_crop_prices.iloc[-1].tolist()
        df_crop_prices_history = df_crop_prices.drop(df_crop_prices.index[-1])

        # Set fixed weights
        weights = [[1.0] * moment_count] * crop_count

        # Set fixed omegas
        omegas = [[1.0] * scenario_count] * crop_count

        # Calculate moments matrix
        moments = []
        for crop_index in range(crop_count):
            temp = [
                (df_crop_prices_history.iloc[:, crop_index] ** (index + 1)).mean()
                for index in range(moment_count)
            ]
            moments.append(
                [
                    temp[0],
```

```
                temp[1] - temp[0] ** 2,
                temp[2] - 3 * temp[0] * temp[1] + 2 * temp[0] ** 3,
                temp[3]
                - 4 * temp[0] * temp[2]
                + 6 * (temp[0] ** 2) * temp[1]
                - 3 * temp[0] ** 4,
            ]
        )

    # Calculate prices
    sample_std = df_crop_prices_history.std(ddof=1).tolist()
    prices = [
        [
            current_prices[crop_index] + stdev_scale_factor * sample_std[crop_index]
            for stdev_scale_factor in stdev_scale_factors
        ]
        for crop_index in range(crop_count)
    ]

    # Calculate covariances
    crop_covariances = df_crop_prices_history.cov()
    covariances = {
        (i, j): crop_covariances.iloc[i, j]
        for i, j in tuple(combinations(range(crop_count), 2))
    }

    return cls(
        weights=weights,
        prices=prices,
        moments=moments,
        covariances=covariances,
        omegas=omegas,
    )


class LpdmpModel(LpModel[LpdmpParams]):
    """LPDMP model.

    This class is a concrete implementation of the LpBase class for the LPDMP model. It
    takes a parameter set LpdmpParams as input.
    """

    def __init__(self, params: LpdmpParams) -> None:
        """Initialize the LpdmpModel class.

        Args:
```

```
        params (LpdmpParams): Parameter set for the LPDMP model.
    """
    super().__init__(params)

def build_problem(self) -> pulp.LpProblem:
    """Build the LPDMP problem.

    Returns:
        pulp.LpProblem: LPDMP problem.
    """
    from math import ceil

    scenario_mid = ceil(self.params.scenario_count / 2) - 1

    # Variables
    variances_plus = pulp.LpVariable.dict(
        "variances_plus",
        (range(self.params.crop_count), range(self.params.moment_count)),
        lowBound=0,
    )
    variances_minus = pulp.LpVariable.dict(
        "variances_minus",
        (range(self.params.crop_count), range(self.params.moment_count)),
        lowBound=0,
    )
    covariances_plus = {
        (i, j): pulp.LpVariable(f"covariances_plus_{i}_{j}", lowBound=0)
        for i, j in self.params.covariances.keys()
    }
    covariances_minus = {
        (i, j): pulp.LpVariable(f"covariances_minus_{i}_{j}", lowBound=0)
        for i, j in self.params.covariances.keys()
    }
    probabilities = pulp.LpVariable.dict(
        "probabilities", range(self.params.scenario_count), lowBound=0
    )
    hat_ECDF = pulp.LpVariable.dict(
        "hat_ECDF",
        (range(self.params.crop_count), range(self.params.scenario_count)),
        lowBound=0,
    )
    deviations_plus = pulp.LpVariable.dict(
        "deviations_plus",
        (range(self.params.crop_count), range(self.params.scenario_count)),
        lowBound=0,
    )
```

```
deviations_minus = pulp.LpVariable.dict(
    "deviations_minus",
    (range(self.params.crop_count), range(self.params.scenario_count)),
    lowBound=0,
)
self.vars.update(
    {
        "variances_plus": variances_plus,
        "variances_minus": variances_minus,
        "covariances_plus": covariances_plus,
        "covariances_minus": covariances_minus,
        "probabilities": probabilities,
        "hat_ECDF": hat_ECDF,
        "deviations_plus": deviations_plus,
        "deviations_minus": deviations_minus,
    }
)

# Objective
problem = pulp.LpProblem("problem", pulp.LpMinimize)
addend_1 = pulp.lpSum(
    self.params.weights[crop_index][moment_index]
    * (
        variances_plus[crop_index, moment_index]
        + variances_minus[crop_index, moment_index]
    )
    for crop_index in range(self.params.crop_count)
    for moment_index in range(1, self.params.moment_count)
)
addend_2 = pulp.lpSum(
    self.params.weights[i][j]
    * (covariances_plus[i, j] + covariances_minus[i, j])
    for i, j in self.params.covariances.keys()
)
addend_3 = pulp.lpSum(
    self.params.omegas[crop_index][scenario_index]
    * (
        deviations_plus[crop_index, scenario_index]
        + deviations_minus[crop_index, scenario_index]
    )
    for crop_index in range(self.params.crop_count)
    for scenario_index in range(self.params.scenario_count)
)
problem += (addend_1 + addend_2 + addend_3, "objective")

# Constraints
```

```
problem += (
    pulp.lpSum(
        probabilities[scenario_index]
        for scenario_index in range(self.params.scenario_count)
    )
    == 1,
    "cons_probabilities_sum",
)
for scenario_index, scenarios in enumerate(zip(*self.params.prices)):
    if any([scenario < 0.0 for scenario in scenarios]):
        problem += (
            probabilities[scenario_index] == 0.0,
            f"cons_probabilities_{scenario_index}",
        )
for crop_index in range(self.params.crop_count):
    problem += (
        pulp.lpSum(
            self.params.prices[crop_index][scenario_index]
            * probabilities[scenario_index]
            for scenario_index in range(self.params.scenario_count)
        )
        + variances_plus[crop_index, 0]
        - variances_minus[crop_index, 0]
        == self.params.moments[crop_index][0],
        f"cons_moment_{crop_index}",
    )
for crop_index in range(self.params.crop_count):
    for moment_index in range(1, self.params.moment_count):
        problem += (
            pulp.lpSum(
                probabilities[scenario_index]
                * (
                    self.params.prices[crop_index][scenario_index]
                    - self.params.moments[crop_index][0]
                )
                ** (moment_index + 1)
                for scenario_index in range(self.params.scenario_count)
            )
            + variances_plus[crop_index, moment_index]
            - variances_minus[crop_index, moment_index]
            == self.params.moments[crop_index][moment_index],
            f"cons_moment_{crop_index}_{moment_index}",
        )
for i, j in self.params.covariances.keys():
    problem += (
        pulp.lpSum(
```

```
                probabilities[scenario_index]
                * (
                    self.params.prices[i][scenario_index]
                    - self.params.moments[i][0]
                )
                * (
                    self.params.prices[j][scenario_index]
                    - self.params.moments[j][0]
                )
                for scenario_index in range(self.params.scenario_count)
            )
            + covariances_plus[i, j]
            - covariances_minus[i, j]
            == self.params.covariances[i, j],
            f"cons_covariances_{i}_{j}",
        )
    for crop_index in range(self.params.crop_count):
        for scenario_index in range(self.params.scenario_count):
            problem += (
                hat_ECDF[crop_index, scenario_index]
                - pulp.lpSum(probabilities[i] for i in range(scenario_index + 1))
                == deviations_plus[crop_index, scenario_index]
                - deviations_minus[crop_index, scenario_index],
                f"deviations_constraint_{crop_index}_{scenario_index}",
            )
    for scenario_index in range(scenario_mid + 1):
        for scenario_index_before in range(scenario_index):
            problem += (
                probabilities[scenario_index]
                >= probabilities[scenario_index_before],
                f"cons_probabilities_{scenario_index}_{scenario_index_before}",
            )
    for scenario_index in range(scenario_mid, self.params.scenario_count):
        for scenario_index_after in range(
            scenario_index + 1, self.params.scenario_count
        ):
            problem += (
                probabilities[scenario_index]
                >= probabilities[scenario_index_after],
                f"cons_probabilities_{scenario_index}_{scenario_index_after}",
            )

    return problem
```

- problems.py

```
from app.solution.data_structures import *
from spfp.sp import (
```

```
    SpModel,
    SpParams,
)
import time

print("Scenario Tree Generation")
print(f"Number of years {year_count}")
print(f"Number of runs {num_runs}")
print("Begin to solve")

final_results = []

for _ in range(num_runs):
    # Read 100 rows data for 3 crops
    df_crop_prices_history = pd.read_csv("../data/crop_prices_history.csv") * 1000

    # Build tree
    storage = Storage(df_crop_prices_history)
    storage.build(year_count)

    # Compute probabilities
    # probabilities[year][scenario]
    probabilities = [[] for _ in range(year_count+1)]
    for year in range(year_count + 1):
        layer = storage.years[year]  # list of nodes == scenarios at this year
        for node in layer:
            scenarioProb = 1
            for p in node.probs:
                scenarioProb *= p
            probabilities[year].append(scenarioProb)

    #  Compute prices_byproduct = [], prices_trader = [], prices_market = []
    #  these are normal list, not data frame
    #  prices_trader[year][crop][trader][scenario] = prices[year][crop][scenario]
    #  prices_market[year][crop][market][scenario] = prices[year][crop][scenario]
    #  prices_byproduct[year][crop][scenario] = prices[year][crop][scenario] * 0.3

    prices_byproduct = []
    prices_trader = []
    prices_market = []
    penalties_demand_trader = []
    penalties_demand_market = []

    for year in range(len(storage.years)):
        year_data_trader = []
        year_data_market = []
```

```
        year_data_byproduct = []
        year_data_penalties_demand_trader = []
        year_data_penalties_demand_market = []

        for crop in range(crop_count):
            crop_data_trader = \
                [[scenario.data.iloc[-1][crop] for scenario in storage.years[year]] for _ in
range(trader_count)]
            crop_data_market = \
                [[scenario.data.iloc[-1][crop] for scenario in storage.years[year]] for _ in
range(market_count)]
            crop_data_byproduct = \
                [scenario.data.iloc[-1][crop] * 0.3 for scenario in storage.years[year]]
            crop_data_penalties_demand_trader = [[scenario.data.iloc[-1][crop]*0.1 for
scenario in storage.years[year]] for _ in range(market_count)]
            crop_data_penalties_demand_market = [[scenario.data.iloc[-1][crop]*0.1 for
scenario in storage.years[year]] for _ in range(market_count)]

            year_data_trader.append(crop_data_trader)
            year_data_market.append(crop_data_market)
            year_data_byproduct.append(crop_data_byproduct)
            year_data_penalties_demand_trader.append(crop_data_penalties_demand_trader)

year_data_penalties_demand_market.append(crop_data_penalties_demand_market)

        prices_trader.append(year_data_trader)
        prices_market.append(year_data_market)
        prices_byproduct.append(year_data_byproduct)
        penalties_demand_trader.append(year_data_penalties_demand_trader)
        penalties_demand_market.append(year_data_penalties_demand_market)

    # Compute expected profit using SPModel
    sp_params = SpParams(
        area_total=area_total,
        costs_water=costs_water,
        costs_light=costs_light,
        costs_harvest=costs_harvest,
        costs_cut=costs_cut,
        costs_process=costs_process,
        costs_worker_fixed=costs_worker_fixed,
        costs_worker_hired=costs_worker_hired,
        amounts_water_required=amounts_water_required,
        amounts_light_required=amounts_light_required,
        probabilities=probabilities,
        areas_initial=areas_initial,
        costs_plant=costs_plant,
```

```
    amounts_demand_byproduct=amounts_demand_byproduct,
    productivities=productivities,
    penalties_demand_trader=penalties_demand_trader,
    penalties_demand_market=penalties_demand_market,
    amounts_demand_trader=amounts_demand_trader,
    amounts_demand_market=amounts_demand_market,
    prices_byproduct=prices_byproduct,
    prices_trader=prices_trader,
    prices_market=prices_market,
    storage=storage,
    count_worker_plant = count_worker_plant,
    count_worker_harvest = count_worker_harvest,
    count_worker_cut = count_worker_cut,
)
sp_model = SpModel(sp_params)


# Record the start time
start_time = time.time()
sp_model.solve()
# Record the end time
end_time = time.time()

# Calculate the elapsed time
elapsed_time = end_time - start_time

# Print the elapsed time
# print(f"Elapsed Time: {elapsed_time} seconds")

c = sp_model.problem.constraints
# print(f"Constraints {len(c)}")
v = sp_model.problem.variables()
# print(f"Variables {len(v)}")
profit = sp_model.problem.objective.value()

# print(f"Expected profit: {sp_model.problem.objective.value()}")
final_results.append(str(profit))

print("Profits")
print(",".join([str(e) for e in final_results]))

#Calculate and display results
results = sp_model.calculate_results()
print(results)
```

- data_structures.py

```
import pandas as pd
import copy
import pickle
from statsmodels.tsa.arima.model import ARIMA

from app.solution.constants import *
from spfp.lpdmp import (
    LpdmpModel,
    LpdmpParams,
)


class Node:
    def __init__(self, probs, data, year):
        self.probs = probs
        self.data = data
        self.year = year


def generate_next(current_node, current_year, result):
    current_data = current_node.data
    # expect that probs is a list of 3 elements, where each is prob of a scenario
    next_value, stds, probs = compute_values(current_data)
    for scenario_index in range(len(STDEV_SCALE_FACTORS)):
        value = next_value + STDEV_SCALE_FACTORS[scenario_index] * stds
        # for dataframe, this is a deep copy
        current_data_copy = current_data
        current_probs_copy = copy.deepcopy(current_node.probs)
        current_probs_copy.append(probs[scenario_index])
        node = Node(current_probs_copy, current_data_copy.append(value),
                    current_year + 1)
        result.append(node)


class Storage:
    def __init__(self, data):
        self.years = [[Node([1], data, 0)]]

    # Build up to number of years
    def build(self, years):
        # load from text files first
        current_year = 1
        while current_year <= years:
            try:
                with open(f"model_{current_year}.txt", "rb") as f:
                    # print(f"Loading year {current_year}")
```

```
                current_layer = pickle.load(f)
                self.years.append(current_layer)
                current_year += 1
        except IOError:
            break
    current_year -= 1
    # loaded current_year. However, need to check if current_year is fully computed
    # expect 3**(current_year) nodes for this layer

    # Check if current_year layer is fully computed
    is_current_layer_full = len(self.years[current_year]) ==
len(STDEV_SCALE_FACTORS) ** current_year

    # compute current layer by previous year
    if not is_current_layer_full:
        current_index = len(self.years[current_year]) - 1
        # locate index of previous node
        index_previous_layer = current_index // len(STDEV_SCALE_FACTORS)
        count = current_index % len(STDEV_SCALE_FACTORS)
        # if previous node already generate full len(STDEV_SCALE_FACTORS), move
to next node
        if count == len(STDEV_SCALE_FACTORS) - 1:
            index_previous_layer += 1
        else:  # otherwise, remove all generated by previous node
            while count >= 0:
                self.years[current_year].pop()
                count -= 1

        while index_previous_layer < len(self.years[current_year - 1]):
            previous_node = self.years[current_year - 1][index_previous_layer]
            generate_next(previous_node, current_year - 1, self.years[current_year])
            # periodically save
            if index_previous_layer % 5 == 0:
                print(f"Saving year {current_year} length {len(self.years[current_year])}")
                with open(f"model_{current_year}.txt", "wb") as f:
                    pickle.dump(self.years[current_year], f)

            index_previous_layer += 1

        with open(f"model_{current_year}.txt", "wb") as f:
            pickle.dump(self.years[current_year], f)

    # compute the rest
    # from current year, build next year
    while current_year < years:
        next_layer = []
```

```
        for current_index, current_node in enumerate(self.years[current_year]):
            generate_next(current_node, current_year, next_layer)
            # periodically save
            if current_index % 5 == 0:
                print(f"Saving year {current_year+1} length {len(next_layer)}")
                with open(f"model_{current_year + 1}.txt", "wb") as f:
                    pickle.dump(next_layer, f)

        # final save
        with open(f"model_{current_year + 1}.txt", "wb") as f:
            pickle.dump(next_layer, f)
        self.years.append(next_layer)
        current_year += 1


# Output: next value, std, probs
def compute_values(data):
    stds = data.std()
    models = [
        ARIMA(price, order=(20, 0, 0)).fit()
        for _, price in data.items()
    ]
    forecast = pd.concat(
        [model.forecast(steps=1) for model in models],
        axis=1,
        keys=data.columns,
    )
    data_copy = data
    data_copy = data_copy.append(
        forecast.loc.obj
    )
    lpdmp_params = LpdmpParams.from_crop_prices_history(
        data_copy, STDEV_SCALE_FACTORS
    )
    lpdmp_model = LpdmpModel(lpdmp_params)
    lpdmp_model.solve()
    probs = [lpdmp_model.vars["probabilities"][scenario_index].varValue
          for scenario_index in range(len(STDEV_SCALE_FACTORS))]

    return forecast.loc.obj, stds, probs



# import pickle
# with open(f"model_{4}.txt", "rb") as f:
```

```
#     current_layer = pickle.load(f)
#     print(1)

# with open(f"model_{5}.txt", "wb") as f:
#     pickle.dump([1,2,3,4,5,6,7,8,9], f)
```

- constants.py

```
STDEV_SCALE_FACTORS = [-1.5, 0, 1.5]
year_count = 8
crop_count = 3
trader_count = 5
market_count = 5
num_runs = 1

area_total = 20
costs_water = 0.1
costs_light = 0.15
costs_harvest = [1700] * (year_count+1)
costs_plant = [[5000] * crop_count] * (year_count+1)
costs_cut = [3000] * (year_count+1)
costs_process = [300] * (year_count+1)
costs_worker_fixed = [5000] * (year_count+1)
costs_worker_hired = [1700] * (year_count+1)
amounts_water_required = [755, 1235, 1925, 2585, 2585, 2585, 2585, 2585, 2585, 2585]
amounts_light_required = [25000] * crop_count
areas_initial = [
    [0, 0, 0, 0, 20, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 20, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
]
productivities = [
    [
        [0, 50, 60, 80, 90, 90, 90, 80, 50, 30],
        [0, 40, 50, 70, 80, 80, 80, 70, 40, 20],
        [0, 10, 20, 30, 30, 30, 30, 20, 10, 10],
    ]
] * (year_count+1)
amounts_demand_byproduct = [[20, 0, 0]] * (year_count+1)
amounts_demand_trader = [
    [[200] * trader_count, [160] * trader_count, [40] * trader_count]
] * (year_count+1)
amounts_demand_market = [
    [[200] * market_count, [160] * market_count, [40] * market_count]
] * (year_count+1)

count_worker_plant = [10,0,0,0,0,0,0,10,0,0]
```

```
count_worker_harvest = [0,3,3,3,3,3,3,3,3,0]
count_worker_cut = [0,0,0,0,0,0,0,6,0,6]
```

APPENDIX C3: PYTHON CODES OF SAMPLE AVERAGE APPROXIMATION METHOD

- sp.py

```
from __future__ import annotations

from dataclasses import (
    dataclass,
    field,
)

import pulp

from lp import (
    LpModel,
    LpParams,
)


@dataclass(frozen=True)
class SpParams(LpParams):
    """Parameter set for the SP model.

    Args:
        crop_count(int): Number of crops.
        age_count(int): Number of ages.
        trader_count(int): Number of traders.
        market_count(int): Number of markets.
        scenario_count(int): Number of scenarios.
        year_count(int): Number of years.
        area_total(int): Total areas of the farm.
        costs_water(float): Costs of water.
        costs_light(float): Costs of light.
        costs_harvest(list[float]): Costs of harvesting, indexed by year.
        costs_cut(list[float]): Costs of cutting, indexed by year.
        costs_process(list[float]): Costs of processing, indexed by year.
        costs_worker_fixed(list[float]): Costs of a fixed worker, indexed by year.
        costs_worker_hired(list[float]): Costs of a hired worker, indexed by year.
        amounts_water_required(list[float]): Amounts of water required, indexed by age.
            Defaults to None.
        amounts_light_required(list[float]): Amounts of light required, indexed by crop.
            Defaults to None.
        probabilities(list[list[float]]): Probabilities of each scenario, indexed by
```

```
        year and scenario.
    areas_initial(list[list[float]]): Initial areas, indexed by crop and age.
    costs_plant(list[list[float]]): Costs of planting, indexed by year and crop.
    amounts_demand_byproduct(list[list[float]]): Amounts of by-products on demand,
        indexed by year and crop.
    productivities(list[list[list[float]]]): Productivities of each crop, indexed by
        year, crop and age.
    penalties_demand_trader(list[list[list[float]]]): Penalties if trade demand is
        not met, indexed by year, crop and trader.
    penalties_demand_market(list[list[list[float]]]): Penalties if market demand is
        not met, indexed by year, crop and market.
    amounts_demand_trader(list[list[list[float]]]): Amounts for trader on demand,
        indexed by year, crop and trader.
    amounts_demand_market(list[list[list[float]]]): Amounts for market on demand,
        indexed by year, crop and market.
    prices_byproduct(list[list[list[float]]]): Prices of by-products, indexed by
        year, crop and scenario.
    prices_trader(list[list[list[list[float]]]]): Prices of traders, indexed by
        year, crop, trader and scenario.
    prices_market(list[list[list[list[float]]]]): Prices of markets, indexed by
        year, crop, market and scenario.
    count_worker_plant(list[int], optional): Number of workers to plant, indexed by
        year. Defaults to None.
    count_worker_harvest(list[int], optional): Number of workers to harvest,
        indexed by year. Defaults to None.
    count_worker_cut(list[int], optional): Number of workers to cut, indexed by
        year. Defaults to None.
    amounts_water_max(float, optional): Maximum amounts of water. Defaults to None.
    amounts_light_max(float, optional): Maximum amounts of light. Defaults to None.
    areas_plant_min(list[list[float]], optional): Minimum areas to plant, indexed
        by year and crop. Defaults to None.

Returns:
    SpParams: Parameter set for the SP model.
"""

crop_count: int = field(init=False)
age_count: int = field(init=False)
trader_count: int = field(init=False)
market_count: int = field(init=False)
scenario_count: int = field(init=False)
year_count: int = field(init=False)
area_total: int
costs_water: float
costs_light: float
costs_harvest: list[float]
```

```python
    costs_cut: list[float]
    costs_process: list[float]
    costs_worker_fixed: list[float]
    costs_worker_hired: list[float]
    amounts_water_required: list[float]
    amounts_light_required: list[float]
    probabilities: list[list[float]]
    areas_initial: list[list[float]]
    costs_plant: list[list[float]]
    amounts_demand_byproduct: list[list[float]]
    productivities: list[list[list[float]]]
    penalties_demand_trader: list[list[list[float]]]
    penalties_demand_market: list[list[list[float]]]
    amounts_demand_trader: list[list[list[float]]]
    amounts_demand_market: list[list[list[float]]]
    prices_byproduct: list[list[list[float]]]
    prices_trader: list[list[list[list[float]]]]
    prices_market: list[list[list[list[float]]]]
    iterations: int
    count_worker_plant: list[float]
    count_worker_harvest: list[float]
    count_worker_cut: list[float]
    areas_plant_min: list[list[float]] = None

    def __post_init__(self) -> None:
        """Post initialization of the SpParams class."""
        object.__setattr__(self, "crop_count", len(self.areas_initial))
        object.__setattr__(self, "age_count", len(self.areas_initial[0]))
        object.__setattr__(self, "trader_count", len(self.prices_trader[0][0]))
        object.__setattr__(self, "market_count", len(self.prices_market[0][0]))
        object.__setattr__(self, "scenario_count", self.iterations)
        object.__setattr__(self, "year_count", len(self.costs_harvest)-1)


class SpModel(LpModel[SpParams]):
    """SP model.

    This class is a concrete implementation of the LpBase class for the SP model. It
    takes a parameter set SpParams as input.
    """

    def __init__(self, params: SpParams) -> None:
        """Initialize the SpModel class.

        Args:
            params (SpParams): Parameter set for the SP model.
```

```
        """
        super().__init__(params)

    def build_problem(self) -> pulp.LpProblem:
        """Build the SP problem.

        Returns:
            pulp.LpProblem: SP problem.
        """
        # Variables
        amounts_trader_shipped = pulp.LpVariable.dict(
            "amounts_trader_shipped",
            (
                range(self.params.year_count+1),
                range(self.params.crop_count),
                range(self.params.trader_count),
                range(self.params.scenario_count),
            ),
            lowBound=0,
        )
        amounts_trader_shipped_under = pulp.LpVariable.dict(
            "amounts_trader_shipped_under",
            (
                range(self.params.year_count+1),
                range(self.params.crop_count),
                range(self.params.trader_count),
                range(self.params.scenario_count),
            ),
            lowBound=0,
        )
        amounts_market_shipped = pulp.LpVariable.dict(
            "amounts_market_shipped",
            (
                range(self.params.year_count+1),
                range(self.params.crop_count),
                range(self.params.market_count),
                range(self.params.scenario_count),
            ),
            lowBound=0,
        )
        amounts_market_shipped_under = pulp.LpVariable.dict(
            "amounts_market_shipped_under",
            (
                range(self.params.year_count+1),
                range(self.params.crop_count),
                range(self.params.market_count),
```

```
        range(self.params.scenario_count),
    ),
    lowBound=0,
)
amounts_byproduct = pulp.LpVariable.dict(
    "amounts_byproduct",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
        range(self.params.scenario_count),
    ),
    lowBound=0,
)
areas = pulp.LpVariable.dict(
    "areas",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
        range(self.params.age_count),
    ),
    lowBound=0,
)
areas_plant = pulp.LpVariable.dict(
    "areas_plant",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
    ),
    lowBound=0,
)
areas_cut = pulp.LpVariable.dict(
    "areas_cut",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
        range(self.params.age_count),
    ),
    lowBound=0,
)
count_worker_fixed = pulp.LpVariable.dict(
    "count_worker_fixed",
    range(self.params.year_count+1),
    lowBound=0,
    cat="Integer",
)
count_worker_hired = pulp.LpVariable.dict(
```

```
    "count_worker_hired",
    range(self.params.year_count+1),
    lowBound=0,
    cat="Integer",
)
self.vars.update(
    {
        "amounts_trader_shipped": amounts_trader_shipped,
        "amounts_trader_shipped_under": amounts_trader_shipped_under,
        "amounts_market_shipped": amounts_market_shipped,
        "amounts_market_shipped_under": amounts_market_shipped_under,
        "amounts_byproduct": amounts_byproduct,
        "areas": areas,
        "areas_plant": areas_plant,
        "areas_cut": areas_cut,
        "count_worker_fixed": count_worker_fixed,
        "count_worker_hired": count_worker_hired,
    }
)

# Objective
problem = pulp.LpProblem("problem", pulp.LpMaximize)
addend_1 = 0

for year in range(1, self.params.year_count + 1):
    for scenario_index in range(self.params.iterations):
        addend_1 += 1 / self.params.iterations * (
            pulp.lpSum(
                self.params.prices_trader[year][crop][trader][scenario_index]
                * amounts_trader_shipped[year, crop, trader, scenario_index]
                for crop in range(self.params.crop_count)
                for trader in range(self.params.trader_count)
            )
            + pulp.lpSum(
                self.params.prices_market[year][crop][market][scenario_index]
                * amounts_market_shipped[year, crop, market, scenario_index]
                for crop in range(self.params.crop_count)
                for market in range(self.params.market_count)
            )
            + pulp.lpSum(
                self.params.prices_byproduct[year][crop][scenario_index]
                * amounts_byproduct[year, crop, scenario_index]
                for crop in range(self.params.crop_count)
            )
            - pulp.lpSum(
                self.params.penalties_demand_trader[year][crop][trader]
```

```
            * amounts_trader_shipped_under[year, crop, trader, scenario_index]
            for crop in range(self.params.crop_count)
            for trader in range(self.params.trader_count)
        )
        - pulp.lpSum(
            self.params.penalties_demand_market[year][crop][market]
            * amounts_market_shipped_under[year, crop, market, scenario_index]
            for crop in range(self.params.crop_count)
            for market in range(self.params.market_count)
        )
        - self.params.costs_process[year]
        * pulp.lpSum(
            amounts_byproduct[year, crop, scenario_index]
            for crop in range(self.params.crop_count)
        )
    )
  addend_2 = -pulp.lpSum(
    self.params.costs_plant[year][crop] * areas_plant[year, crop]
    for crop in range(self.params.crop_count)
    for age in range(self.params.age_count)
  )
  addend_3 = -self.params.costs_harvest[year] * pulp.lpSum(
    areas[year, crop, age]
    for crop in range(self.params.crop_count)
    for age in range(self.params.age_count)
  )
  addend_4 = -self.params.costs_cut[year] * pulp.lpSum(
    areas_cut[year, crop, age]
    for crop in range(self.params.crop_count)
    for age in range(self.params.age_count)
  )
  addend_5 = -self.params.costs_worker_fixed[year] * count_worker_fixed[year]
  addend_6 = -self.params.costs_worker_hired[year] * count_worker_hired[year]
  addend_7 = -self.params.costs_water * pulp.lpSum(
    self.params.amounts_water_required[age] * areas[year, crop, age]
    for crop in range(self.params.crop_count)
    for age in range(self.params.age_count)
  )
  addend_8 = -self.params.costs_light * pulp.lpSum(
    self.params.amounts_light_required[crop] * areas[year, crop, age]
    for crop in range(self.params.crop_count)
    for age in range(self.params.age_count)
  )
problem += (
  addend_1
  + addend_2
```

```
    + addend_3
    + addend_4
    + addend_5
    + addend_6
    + addend_7
    + addend_8,
    "objective",
)

# Constraints
for year in range(1, self.params.year_count+1):
    problem += (
        pulp.lpSum(
            areas[year, crop, age]
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )
        <= self.params.area_total,
        f"cons_areas_used_year_{year}",
    )
    for crop in range(self.params.crop_count):
        problem += (
            areas[year, crop, 0] == areas_plant[year, crop],
            f"cons_areas_crop_{crop}_age_0_year_{year}",
        )
    for crop in range(self.params.crop_count):
        problem += (
            areas_cut[year, crop, 0] == 0,
            f"cons_areas_cut_crop_{crop}_age_first_year_{year}",
        )
    if year == 1:
        for crop in range(self.params.crop_count):
            for age in range(1, self.params.age_count):
                problem += (
                    areas[year, crop, age]
                    == self.params.areas_initial[crop][age - 1]
                    - areas_cut[year, crop, age],
                    f"cons_areas_crop_{crop}_age_{age}_year_{year}",
                )
        for crop in range(self.params.crop_count):
            problem += (
                areas_cut[year, crop, self.params.age_count - 1]
                == self.params.areas_initial[crop][
                    self.params.age_count - 1 - 1
                ],
                f"cons_areas_cut_crop_{crop}_age_last_year_{year}",
```

```
        )
else:
    for crop in range(self.params.crop_count):
        for age in range(1, self.params.age_count):
            problem += (
                areas[year, crop, age]
                == areas[year - 1, crop, age - 1]
                - areas_cut[year, crop, age],
                f"cons_areas_crop_{crop}_age_{age}_year_{year}",
            )
    for crop in range(self.params.crop_count):
        problem += (
            areas_cut[year, crop, self.params.age_count - 1]
            == areas[year - 1, crop, self.params.age_count - 1 - 1],
            f"cons_areas_cut_crop_{crop}_age_last_year_{year}",
        )
for crop in range(self.params.crop_count):
    for scenario in range(self.params.scenario_count):
        problem += (
            pulp.lpSum(
                amounts_trader_shipped[year, crop, trader, scenario]
                for trader in range(self.params.trader_count)
            )
            + pulp.lpSum(
                amounts_market_shipped[year, crop, market, scenario]
                for market in range(self.params.market_count)
            )
            + amounts_byproduct[year, crop, scenario]
            <= pulp.lpSum(
                self.params.productivities[year][crop][age]
                * areas[year, crop, age]
                for age in range(self.params.age_count)
            ),
            "cons_productivities_"
            f"crop_{crop}_scenario_{scenario}_year_{year}",
        )
for crop in range(self.params.crop_count):
    for trader in range(self.params.trader_count):
        for scenario in range(self.params.scenario_count):
            problem += (
                amounts_trader_shipped[year, crop, trader, scenario]
                == self.params.amounts_demand_trader[year][crop][trader]
                - amounts_trader_shipped_under[
                    year, crop, trader, scenario
                ],
                "cons_amounts_"
```

```
                f"crop_{crop}_"
                f"trader_{trader}_"
                f"scenario_{scenario}_"
                f"year_{year}",
            )
for crop in range(self.params.crop_count):
    for market in range(self.params.market_count):
        for scenario in range(self.params.scenario_count):
            problem += (
                amounts_market_shipped[year, crop, market, scenario]
                == self.params.amounts_demand_market[year][crop][market]
                - amounts_market_shipped_under[
                    year, crop, market, scenario
                ],
                "cons_amounts_"
                f"crop_{crop}_"
                f"market_{market}_"
                f"scenario_{scenario}_"
                f"year_{year}",
            )
for crop in range(self.params.crop_count):
    for scenario in range(self.params.scenario_count):
        problem += (
            amounts_byproduct[year, crop, scenario]
            <= self.params.amounts_demand_byproduct[year][crop],
            "cons_amounts_byproduct_"
            f"crop_{crop}_scenario_{scenario}_year_{year}",
        )
problem += (
    count_worker_fixed[year]
    <= 5,
    "cons_count_worker_fixed_"
    f"year_{year}",
)
problem += (
    count_worker_hired[year]
    <= 1000,
    "cons_count_worker_hired_"
    f"year_{year}",
)
total_worker_area_plant = pulp.lpSum(
    self.params.count_worker_plant[year] *
    areas_plant[year, crop]
    for crop in range(self.params.crop_count)
)
total_worker_area_harvest = pulp.lpSum(
```

```
            self.params.count_worker_harvest[year] *
            areas[year, crop, age]
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )
        total_worker_area_cut = pulp.lpSum(
            self.params.count_worker_cut[year] *
            areas_cut[year, crop, age]
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )
        problem += (
            count_worker_fixed[year] + count_worker_hired[year] ==
            total_worker_area_plant +
            total_worker_area_harvest +
            total_worker_area_cut,
            "cons_labour_requirement_"
            f"year_{year}",
        )
    return problem

def calculate_results(self):
    # Retrieve the total profit from the solved objective value
    total_profit = self.problem.objective.value()

    # Calculate total cost including penalties and total yield
    total_cost = 0
    total_yield = 0

    for year in range(1, self.params.year_count + 1):
        # Accumulate total cost including penalties
        total_cost += (
            # addend_1
            sum(
            self.params.penalties_demand_trader[year][crop][trader] *
            self.vars["amounts_trader_shipped_under"][year,        crop,        trader,
scenario_index].varValue
            for crop in range(self.params.crop_count)
            for trader in range(self.params.trader_count)
            for scenario_index in range(self.params.iterations)
            )
            + sum(self.params.penalties_demand_market[year][crop][market] *
                self.vars["amounts_market_shipped_under"][year,        crop,        market,
scenario_index].varValue
                for crop in range(self.params.crop_count)
                for market in range(self.params.market_count)
```

```
        for scenario_index in range(self.params.iterations)
    )
    + self.params.costs_process[year] * sum(
        self.vars["amounts_byproduct"][year, crop,scenario_index].varValue
        for crop in range(self.params.crop_count)
        for scenario_index in range(self.params.iterations)
    )
    # addend_2
    + sum(self.params.costs_plant[year][crop] * self.vars["areas_plant"][year,
crop].varValue
            for crop in range(self.params.crop_count)
    )
    # addend_3
    + self.params.costs_harvest[year] * sum(
        self.vars["areas"][year, crop, age].varValue
        for crop in range(self.params.crop_count)
        for age in range(self.params.age_count)
    )
    # addend_4
    + self.params.costs_cut[year] * sum(
        self.vars["areas_cut"][year, crop, age].varValue
        for crop in range(self.params.crop_count)
        for age in range(self.params.age_count)
    )
    # addend_5
    + self.params.costs_worker_fixed[year] *
self.vars["count_worker_fixed"][year].varValue

    # addend_6
    + self.params.costs_worker_hired[year] *
self.vars["count_worker_hired"][year].varValue

    # addend_7
    + self.params.costs_water * sum(
        self.params.amounts_water_required[age] * self.vars["areas"][year, crop,
age].varValue
        for crop in range(self.params.crop_count)
        for age in range(self.params.age_count)
    )
    # addend_8
    + self.params.costs_light * sum(
        self.params.amounts_light_required[crop] * self.vars["areas"][year, crop,
age].varValue
        for crop in range(self.params.crop_count)
        for age in range(self.params.age_count)
    )
```

```
        )

        total_yield += sum(
            self.params.productivities[year][crop][age] * self.vars["areas"][year, crop, age].varValue
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )

    # Calculating the derived metrics
    average_yield_per_hectare_per_year = total_yield / (self.params.area_total * self.params.year_count) if total_yield > 0 else 0
    average_cost_per_hectare_per_year = total_cost / (self.params.area_total * self.params.year_count) if total_yield > 0 else 0
    average_cost_per_kg_per_year = total_cost / (total_yield * 1000) if total_yield > 0 else 0
    average_profit_per_hectare_per_year = total_profit / (self.params.area_total * self.params.year_count) if total_yield > 0 else 0
    average_profit_per_kg_per_year = total_profit / (total_yield * 1000) if total_yield > 0 else 0

    # Return results in a dictionary
    print("Total profit:", total_profit)
    print("Total cost:", total_cost)
    print("Total yield (tons):", total_yield)
    print("Average yield per hectare per year (tons/ha/year):", average_yield_per_hectare_per_year)
    print("Average cost per hectare per year:", average_cost_per_hectare_per_year)
    print("Average cost per kilogram per year:", average_cost_per_kg_per_year)
    print("Average profit per hectare per year:", average_profit_per_hectare_per_year)
    print("Average profit per kilogram per year:", average_profit_per_kg_per_year)
```

- problems.py

```
from constants import *
import pandas as pd
from scipy.stats import norm
import random
import time
from sp import (
  SpModel,
  SpParams,
)

# Read 100 rows data for 3 crops
df_crop_prices_history = pd.read_csv("../data/crop_prices_history.csv") * 1000
```

```
# each crop, compute mean and std
df_crop_mean = [df_crop_prices_history.iloc[:, i].mean() for i in
range(len(df_crop_prices_history.columns))]
df_crop_std = [df_crop_prices_history.iloc[:, i].std() for i in
range(len(df_crop_prices_history.columns))]

# Compute probabilities
# probabilities[year][scenario]
probabilities = [[1 / iterations for i in range(iterations)] for _ in range(year_count + 1)]

def normal_dist_random(mean, std):
   return mean + norm.ppf(random.random()) * std

print('SAA')
print(f"Number of years: {year_count}")
print(f"Number of iterations {iterations}")
print(f"Number of runs {num_runs}")
records = []

#  Compute prices_byproduct = [], prices_trader = [], prices_market = []
#  these are normal list, not data frame
#  prices_trader[year][crop][trader][scenario] = prices[year][crop][scenario]
#  prices_market[year][crop][market][scenario] = prices[year][crop][scenario]
#  prices_byproduct[year][crop][scenario] = prices[year][crop][scenario] * 0.3

finalResults = []
for indexSpecial in range(num_runs):
   prices_byproduct = []
   prices_trader = []
   prices_market = []
   penalties_demand_trader = []
   penalties_demand_market = []

   for year in range(year_count + 1):
      year_data_trader = []
      year_data_market = []
      year_data_byproduct = []
      year_data_penalties_demand_trader = []
      year_data_penalties_demand_market = []

      for crop in range(crop_count):
         prices = [normal_dist_random(df_crop_mean[crop], df_crop_std[crop]) for
scenario in range(iterations)]
         crop_data_trader = [prices for _ in range(trader_count)]
         crop_data_market = [prices for _ in range(market_count)]
```

```
            crop_data_byproduct = [p*0.3 for p in prices]
            crop_data_penalties_demand_trader = [p*0.1 for p in prices]
            crop_data_penalties_demand_market = [p*0.1 for p in prices]

            year_data_trader.append(crop_data_trader)
            year_data_market.append(crop_data_market)
            year_data_byproduct.append(crop_data_byproduct)
            year_data_penalties_demand_trader.append(crop_data_penalties_demand_trader)

year_data_penalties_demand_market.append(crop_data_penalties_demand_market)

        prices_trader.append(year_data_trader)
        prices_market.append(year_data_market)
        prices_byproduct.append(year_data_byproduct)
        penalties_demand_trader.append(year_data_penalties_demand_trader)
        penalties_demand_market.append(year_data_penalties_demand_market)

    # Compute expected profit using SPModel
    sp_params = SpParams(
        area_total=area_total,
        costs_water=costs_water,
        costs_light=costs_light,
        costs_harvest=costs_harvest,
        costs_cut=costs_cut,
        costs_process=costs_process,
        costs_worker_fixed=costs_worker_fixed,
        costs_worker_hired=costs_worker_hired,
        amounts_water_required=amounts_water_required,
        amounts_light_required=amounts_light_required,
        probabilities=probabilities,
        areas_initial=areas_initial,
        costs_plant=costs_plant,
        amounts_demand_byproduct=amounts_demand_byproduct,
        productivities=productivities,
        penalties_demand_trader=penalties_demand_trader,
        penalties_demand_market=penalties_demand_market,
        amounts_demand_trader=amounts_demand_trader,
        amounts_demand_market=amounts_demand_market,
        prices_byproduct=prices_byproduct,
        prices_trader=prices_trader,
        prices_market=prices_market,
        iterations=iterations,
        count_worker_plant=count_worker_plant,
        count_worker_harvest=count_worker_harvest,
        count_worker_cut=count_worker_cut,
    )
```

```
    sp_model = SpModel(sp_params)
    # print("Begin to solve")
    # Record the start time
    start_time = time.time()
    sp_model.solve()
    # Record the end time
    end_time = time.time()

    # Calculate the elapsed time
    elapsed_time = end_time - start_time

    # Print the elapsed time
    # print(f"Elapsed Time: {elapsed_time} seconds")

    c = sp_model.problem.constraints
    # print(f"Constraints {len(c)}")
    v = sp_model.problem.variables()
    # print(f"Variables {len(v)}")

    profit = sp_model.problem.objective.value()
    # print(f"Profit: {profit}")
    finalResults.append(profit)

# print("Profits")
# print(",".join([str(e) for e in finalResults]))

# Calculate and display results
results = sp_model.calculate_results()
print(results)
```

- constants.py

```
STDEV_SCALE_FACTORS = [-1.5, 0, 1.5]
iterations = 6561
# 2000,4000,6000
year_count = 8
crop_count = 3
trader_count = 5
market_count = 5
num_runs = 1

area_total = 20
costs_water = 0.1
costs_light = 0.15
costs_harvest = [1700] * (year_count+1)
costs_plant = [[5000] * crop_count] * (year_count+1)
costs_cut = [3000] * (year_count+1)
```

```
costs_process = [300] * (year_count+1)
costs_worker_fixed = [5000] * (year_count+1)
costs_worker_hired = [1700] * (year_count+1)
amounts_water_required = [755, 1235, 1925, 2585, 2585, 2585, 2585, 2585, 2585, 2585]
amounts_light_required = [25000] * crop_count
areas_initial = [
    [0, 0, 0, 0, 20, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 20, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
]
productivities = [
    [
        [0, 50, 60, 80, 90, 90, 90, 80, 50, 30],
        [0, 40, 50, 70, 80, 80, 80, 70, 40, 20],
        [0, 10, 20, 30, 30, 30, 30, 20, 10, 10],
    ]
] * (year_count+1)
amounts_demand_byproduct = [[20, 0, 0]] * (year_count+1)
amounts_demand_trader = [
    [[200] * trader_count, [160] * trader_count, [40] * trader_count]
] * (year_count+1)
amounts_demand_market = [
    [[200] * market_count, [160] * market_count, [40] * market_count]
] * (year_count+1)

count_worker_plant = [10,0,0,0,0,0,0,10,0,0]

count_worker_harvest = [0,3,3,3,3,3,3,3,3,0]

count_worker_cut = [0,0,0,0,0,0,0,6,0,6]
```

APPENDIX C4: PYTHON CODES OF CHANCE CONSTRAINED PROGRAMMING APPROACH

- sp.py

```
from __future__ import annotations

from dataclasses import (
    dataclass,
    field,
)

import pulp

from lp import (
    LpModel,
    LpParams,
)


@dataclass(frozen=True)
class SpParams(LpParams):
    """Parameter set for the SP model.

    Args:
        crop_count(int): Number of crops.
        age_count(int): Number of ages.
        trader_count(int): Number of traders.
        market_count(int): Number of markets.
        scenario_count(int): Number of scenarios.
        year_count(int): Number of years.
        area_total(int): Total areas of the farm.
        costs_water(float): Costs of water.
        costs_light(float): Costs of light.
        costs_harvest(list[float]): Costs of harvesting, indexed by year.
        costs_cut(list[float]): Costs of cutting, indexed by year.
        costs_process(list[float]): Costs of processing, indexed by year.
        costs_worker_fixed(list[float]): Costs of a fixed worker, indexed by year.
        costs_worker_hired(list[float]): Costs of a hired worker, indexed by year.
        amounts_water_required(list[float]): Amounts of water required, indexed by age.
            Defaults to None.
        amounts_light_required(list[float]): Amounts of light required, indexed by crop.
            Defaults to None.
        areas_initial(list[list[float]]): Initial areas, indexed by crop and age.
```

```
    costs_plant(list[list[float]]): Costs of planting, indexed by year and crop.
    amounts_demand_byproduct(list[list[float]]): Amounts of by-products on demand,
        indexed by year and crop.
    productivities(list[list[list[float]]]): Productivities of each crop, indexed by
        year, crop and age.
    penalties_demand_trader(list[list[list[float]]]): Penalties if trade demand is
        not met, indexed by year, crop and trader.
    penalties_demand_market(list[list[list[float]]]): Penalties if market demand is
        not met, indexed by year, crop and market.
    amounts_demand_trader(list[list[list[float]]]): Amounts for trader on demand,
        indexed by year, crop and trader.
    amounts_demand_market(list[list[list[float]]]): Amounts for market on demand,
        indexed by year, crop and market.
    prices_byproduct(list[list[list[float]]]): Prices of by-products, indexed by
        year, crop and scenario.
    prices_trader(list[list[list[list[float]]]]): Prices of traders, indexed by
        year, crop, trader and scenario.
    prices_market(list[list[list[list[float]]]]): Prices of markets, indexed by
        year, crop, market and scenario.
    count_worker_plant(list[int], optional): Number of workers to plant, indexed by
        year. Defaults to None.
    count_worker_harvest(list[int], optional): Number of workers to harvest,
        indexed by year. Defaults to None.
    count_worker_cut(list[int], optional): Number of workers to cut, indexed by
        year. Defaults to None.
    amounts_water_max(float, optional): Maximum amounts of water. Defaults to None.
    amounts_light_max(float, optional): Maximum amounts of light. Defaults to None.
    areas_plant_min(list[list[float]], optional): Minimum areas to plant, indexed
        by year and crop. Defaults to None.

Returns:
    SpParams: Parameter set for the SP model.
"""

crop_count: int = field(init=False)
age_count: int = field(init=False)
trader_count: int = field(init=False)
market_count: int = field(init=False)
scenario_count: int = field(init=False)
year_count: int = field(init=False)
area_total: int
costs_water: float
costs_light: float
costs_harvest: list[float]
costs_cut: list[float]
costs_process: list[float]
```

```
    costs_worker_fixed: list[float]
    costs_worker_hired: list[float]
    amounts_water_required: list[float]
    amounts_light_required: list[float]
    areas_initial: list[list[float]]
    costs_plant: list[list[float]]
    amounts_demand_byproduct: list[list[float]]
    productivities: list[list[list[float]]]
    penalties_demand_trader: list[list[list[float]]]
    penalties_demand_market: list[list[list[float]]]
    amounts_demand_trader: list[list[list[float]]]
    amounts_demand_market: list[list[list[float]]]
    prices_byproduct: list[list[list[float]]]
    prices_trader: list[list[list[list[float]]]]
    prices_market: list[list[list[list[float]]]]
    iterations: int
    desired_reliability_trader: float
    desired_reliability_market: float
    count_worker_plant: list[float]
    count_worker_harvest: list[float]
    count_worker_cut: list[float]
    areas_plant_min: list[list[float]] = None

    def __post_init__(self) -> None:
        """Post initialization of the SpParams class."""
        object.__setattr__(self, "crop_count", len(self.areas_initial))
        object.__setattr__(self, "age_count", len(self.areas_initial[0]))
        object.__setattr__(self, "trader_count", len(self.prices_trader[0][0]))
        object.__setattr__(self, "market_count", len(self.prices_market[0][0]))
        object.__setattr__(self, "scenario_count", self.iterations)
        object.__setattr__(self, "year_count", len(self.costs_harvest)-1)


class SpModel(LpModel[SpParams]):
    """SP model.

    This class is a concrete implementation of the LpBase class for the SP model. It
    takes a parameter set SpParams as input.
    """

    def __init__(self, params: SpParams) -> None:
        """Initialize the SpModel class.

        Args:
            params (SpParams): Parameter set for the SP model.
        """
```

```
        super().__init__(params)

    def build_problem(self) -> pulp.LpProblem:
        """Build the SP problem.

        Returns:
            pulp.LpProblem: SP problem.
        """
        # Variables
        binary_trader_shipped_under = pulp.LpVariable.dict(
            "binary_trader_shipped_under",
            (
                range(self.params.year_count + 1),
                range(self.params.crop_count),
                range(self.params.trader_count),
                range(self.params.scenario_count),
            ),
            cat="Binary",
        )
        binary_market_shipped_under = pulp.LpVariable.dict(
            "binary_market_shipped_under",
            (
                range(self.params.year_count + 1),
                range(self.params.crop_count),
                range(self.params.market_count),
                range(self.params.scenario_count),
            ),
            cat="Binary",
        )
        amounts_trader_shipped = pulp.LpVariable.dict(
            "amounts_trader_shipped",
            (
                range(self.params.year_count+1),
                range(self.params.crop_count),
                range(self.params.trader_count),
                range(self.params.scenario_count),
            ),
            lowBound=0,
        )
        amounts_trader_shipped_under = pulp.LpVariable.dict(
            "amounts_trader_shipped_under",
            (
                range(self.params.year_count+1),
                range(self.params.crop_count),
                range(self.params.trader_count),
                range(self.params.scenario_count),
```

```
    ),
    lowBound=0,
)
amounts_market_shipped = pulp.LpVariable.dict(
    "amounts_market_shipped",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
        range(self.params.market_count),
        range(self.params.scenario_count),
    ),
    lowBound=0,
)
amounts_market_shipped_under = pulp.LpVariable.dict(
    "amounts_market_shipped_under",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
        range(self.params.market_count),
        range(self.params.scenario_count),
    ),
    lowBound=0,
)
amounts_byproduct = pulp.LpVariable.dict(
    "amounts_byproduct",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
        range(self.params.scenario_count),
    ),
    lowBound=0,
)
areas = pulp.LpVariable.dict(
    "areas",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
        range(self.params.age_count),
    ),
    lowBound=0,
)
areas_plant = pulp.LpVariable.dict(
    "areas_plant",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
```

```
    ),
    lowBound=0,
)
areas_cut = pulp.LpVariable.dict(
    "areas_cut",
    (
        range(self.params.year_count+1),
        range(self.params.crop_count),
        range(self.params.age_count),
    ),
    lowBound=0,
)
count_worker_fixed = pulp.LpVariable.dict(
    "count_worker_fixed",
    range(self.params.year_count+1),
    lowBound=0,
    cat="Integer",
)
count_worker_hired = pulp.LpVariable.dict(
    "count_worker_hired",
    range(self.params.year_count+1),
    lowBound=0,
    cat="Integer",
)
self.vars.update(
    {
        "amounts_trader_shipped": amounts_trader_shipped,
        "amounts_trader_shipped_under": amounts_trader_shipped_under,
        "amounts_market_shipped": amounts_market_shipped,
        "amounts_market_shipped_under": amounts_market_shipped_under,
        "amounts_byproduct": amounts_byproduct,
        "areas": areas,
        "areas_plant": areas_plant,
        "areas_cut": areas_cut,
        "count_worker_fixed": count_worker_fixed,
        "count_worker_hired": count_worker_hired,
        "binary_trader_shipped_under": binary_trader_shipped_under,
        "binary_market_shipped_under": binary_market_shipped_under,
    }
)

# Objective
problem = pulp.LpProblem("problem", pulp.LpMaximize)
addend_1 = 0

for year in range(1, self.params.year_count + 1):
```

```
for scenario_index in range(self.params.iterations):
    addend_1 += 1 / self.params.iterations * (
        pulp.lpSum(
            self.params.prices_trader[year][crop][trader][scenario_index]
            * amounts_trader_shipped[year, crop, trader, scenario_index]
            for crop in range(self.params.crop_count)
            for trader in range(self.params.trader_count)
        )
        + pulp.lpSum(
            self.params.prices_market[year][crop][market][scenario_index]
            * amounts_market_shipped[year, crop, market, scenario_index]
            for crop in range(self.params.crop_count)
            for market in range(self.params.market_count)
        )
        + pulp.lpSum(
            self.params.prices_byproduct[year][crop][scenario_index]
            * amounts_byproduct[year, crop, scenario_index]
            for crop in range(self.params.crop_count)
        )
        # - pulp.lpSum(
        #     self.params.penalties_demand_trader[year][crop][trader]
        #     * amounts_trader_shipped_under[year, crop, trader, scenario_index]
        #     for crop in range(self.params.crop_count)
        #     for trader in range(self.params.trader_count)
        # )
        # - pulp.lpSum(
        #     self.params.penalties_demand_market[year][crop][market]
        #     * amounts_market_shipped_under[year, crop, market, scenario_index]
        #     for crop in range(self.params.crop_count)
        #     for market in range(self.params.market_count)
        # )
        - self.params.costs_process[year]
        * pulp.lpSum(
            amounts_byproduct[year, crop, scenario_index]
            for crop in range(self.params.crop_count)
        )
    )
addend_2 = -pulp.lpSum(
    self.params.costs_plant[year][crop] * areas_plant[year, crop]
    for crop in range(self.params.crop_count)
    for age in range(self.params.age_count)
)
addend_3 = -self.params.costs_harvest[year] * pulp.lpSum(
    areas[year, crop, age]
    for crop in range(self.params.crop_count)
    for age in range(self.params.age_count)
```

```
    )
    addend_4 = -self.params.costs_cut[year] * pulp.lpSum(
        areas_cut[year, crop, age]
        for crop in range(self.params.crop_count)
        for age in range(self.params.age_count)
    )
    addend_5 = -self.params.costs_worker_fixed[year] * count_worker_fixed[year]
    addend_6 = -self.params.costs_worker_hired[year] * count_worker_hired[year]
    addend_7 = -self.params.costs_water * pulp.lpSum(
        self.params.amounts_water_required[age] * areas[year, crop, age]
        for crop in range(self.params.crop_count)
        for age in range(self.params.age_count)
    )
    addend_8 = -self.params.costs_light * pulp.lpSum(
        self.params.amounts_light_required[crop] * areas[year, crop, age]
        for crop in range(self.params.crop_count)
        for age in range(self.params.age_count)
    )
problem += (
    addend_1
    + addend_2
    + addend_3
    + addend_4
    + addend_5
    + addend_6
    + addend_7
    + addend_8,
    "objective",
)

# Constraints
for year in range(1, self.params.year_count+1):
    problem += (
        pulp.lpSum(
            areas[year, crop, age]
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )
        <= self.params.area_total,
        f"cons_areas_used_year_{year}",
    )
    for crop in range(self.params.crop_count):
        problem += (
            areas[year, crop, 0] == areas_plant[year, crop],
            f"cons_areas_crop_{crop}_age_0_year_{year}",
        )
```

```
for crop in range(self.params.crop_count):
    problem += (
        areas_cut[year, crop, 0] == 0,
        f"cons_areas_cut_crop_{crop}_age_first_year_{year}",
    )
if year == 1:
    for crop in range(self.params.crop_count):
        for age in range(1, self.params.age_count):
            problem += (
                areas[year, crop, age]
                == self.params.areas_initial[crop][age - 1]
                - areas_cut[year, crop, age],
                f"cons_areas_crop_{crop}_age_{age}_year_{year}",
            )
    for crop in range(self.params.crop_count):
        problem += (
            areas_cut[year, crop, self.params.age_count - 1]
            == self.params.areas_initial[crop][
                self.params.age_count - 1 - 1
            ],
            f"cons_areas_cut_crop_{crop}_age_last_year_{year}",
        )
else:
    for crop in range(self.params.crop_count):
        for age in range(1, self.params.age_count):
            problem += (
                areas[year, crop, age]
                == areas[year - 1, crop, age - 1]
                - areas_cut[year, crop, age],
                f"cons_areas_crop_{crop}_age_{age}_year_{year}",
            )
    for crop in range(self.params.crop_count):
        problem += (
            areas_cut[year, crop, self.params.age_count - 1]
            == areas[year - 1, crop, self.params.age_count - 1 - 1],
            f"cons_areas_cut_crop_{crop}_age_last_year_{year}",
        )
for crop in range(self.params.crop_count):
    for scenario in range(self.params.scenario_count):
        problem += (
            pulp.lpSum(
                amounts_trader_shipped[year, crop, trader, scenario]
                for trader in range(self.params.trader_count)
            )
            + pulp.lpSum(
                amounts_market_shipped[year, crop, market, scenario]
```

```
                for market in range(self.params.market_count)
            )
            + amounts_byproduct[year, crop, scenario]
            <= pulp.lpSum(
                self.params.productivities[year][crop][age]
                * areas[year, crop, age]
                for age in range(self.params.age_count)
            ),
            "cons_productivities_"
            f"crop_{crop}_scenario_{scenario}_year_{year}",
        )
for crop in range(self.params.crop_count):
    for trader in range(self.params.trader_count):
        for scenario in range(self.params.scenario_count):
            # problem += (
            #     amounts_trader_shipped[year, crop, trader, scenario]
            #     == self.params.amounts_demand_trader[year][crop][trader]
            #     - amounts_trader_shipped_under[
            #         year, crop, trader, scenario
            #     ],
            #     "cons_amounts_"
            #     f"crop_{crop}_"
            #     f"trader_{trader}_"
            #     f"scenario_{scenario}_"
            #     f"year_{year}",
            # )

            problem += (
                self.params.amounts_demand_trader[year][crop][trader] -
                amounts_trader_shipped[year, crop, trader, scenario] ==
                5000 * binary_trader_shipped_under[year, crop, trader, scenario]
            )

for crop in range(self.params.crop_count):
    for market in range(self.params.market_count):
        for scenario in range(self.params.scenario_count):
            # problem += (
            #     amounts_market_shipped[year, crop, market, scenario]
            #     == self.params.amounts_demand_market[year][crop][market]
            #     - amounts_market_shipped_under[
            #         year, crop, market, scenario
            #     ],
            #     "cons_amounts_"
            #     f"crop_{crop}_"
            #     f"market_{market}_"
            #     f"scenario_{scenario}_"
```

```
                #     f"year_{year}",
                # )

                problem += (
                    self.params.amounts_demand_market[year][crop][market] -
                    amounts_market_shipped[year, crop, market, scenario] ==
                    5000 * binary_market_shipped_under[year, crop, market, scenario]
                )

        for crop in range(self.params.crop_count):
            for scenario in range(self.params.scenario_count):
                problem += (
                    amounts_byproduct[year, crop, scenario]
                    <= self.params.amounts_demand_byproduct[year][crop],
                    "cons_amounts_byproduct_"
                    f"crop_{crop}_scenario_{scenario}_year_{year}",
                )

        for crop in range(self.params.crop_count):
            for trader in range(self.params.trader_count):
                for scenario in range(self.params.scenario_count):
                    problem += (
                        pulp.lpSum(
                            1 / self.params.scenario_count * binary_trader_shipped_under[year,
crop, trader, scenario]
                        )
                        <= 1 - self.params.desired_reliability_trader
                    )

        for crop in range(self.params.crop_count):
            for market in range(self.params.market_count):
                for scenario in range(self.params.scenario_count):
                    problem += (
                        pulp.lpSum(
                            1 / self.params.scenario_count * binary_market_shipped_under[year,
crop, market, scenario]
                        )
                        <= 1 - self.params.desired_reliability_market
                    )
        problem += (
            count_worker_fixed[year]
            <= 5,
            "cons_count_worker_fixed_"
            f"year_{year}",
        )
        problem += (
```

```
            count_worker_hired[year]
            <= 1000,
            "cons_count_worker_hired_"
            f"year_{year}",
        )
        total_worker_area_plant = pulp.lpSum(
            self.params.count_worker_plant[year] *
            areas_plant[year, crop]
            for crop in range(self.params.crop_count)
        )
        total_worker_area_harvest = pulp.lpSum(
            self.params.count_worker_harvest[year] *
            areas[year, crop, age]
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )
        total_worker_area_cut = pulp.lpSum(
            self.params.count_worker_cut[year] *
            areas_cut[year, crop, age]
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )
        problem += (
            count_worker_fixed[year] + count_worker_hired[year] ==
            total_worker_area_plant +
            total_worker_area_harvest +
            total_worker_area_cut,
            "cons_labour_requirement_"
            f"year_{year}",
        )
    return problem

def calculate_results(self):
    # Retrieve the total profit from the solved objective value
    total_profit = self.problem.objective.value()

    # Calculate total cost including penalties and total yield
    total_cost = 0
    total_yield = 0

    for year in range(1, self.params.year_count + 1):
        # Accumulate total cost including penalties
        total_cost += (
            # addend_1
            self.params.costs_process[year] * sum(
                self.vars["amounts_byproduct"][year, crop,scenario_index].varValue
```

```
            for crop in range(self.params.crop_count)
            for scenario_index in range(self.params.iterations)
        )
        # addend_2
        + sum(self.params.costs_plant[year][crop] * self.vars["areas_plant"][year,
crop].varValue
                for crop in range(self.params.crop_count)
        )
        # addend_3
        + self.params.costs_harvest[year] * sum(
            self.vars["areas"][year, crop, age].varValue
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )
        # addend_4
        + self.params.costs_cut[year] * sum(
            self.vars["areas_cut"][year, crop, age].varValue
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )
        # addend_5
        + self.params.costs_worker_fixed[year] *
self.vars["count_worker_fixed"][year].varValue

        # addend_6
        + self.params.costs_worker_hired[year] *
self.vars["count_worker_hired"][year].varValue

        # addend_7
        + self.params.costs_water * sum(
            self.params.amounts_water_required[age] * self.vars["areas"][year, crop,
age].varValue
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )
        # addend_8
        + self.params.costs_light * sum(
            self.params.amounts_light_required[crop] * self.vars["areas"][year, crop,
age].varValue
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )

    )

    total_yield += sum(
```

```
            self.params.productivities[year][crop][age] * self.vars["areas"][year, crop,
age].varValue
            for crop in range(self.params.crop_count)
            for age in range(self.params.age_count)
        )

    # Calculating the derived metrics
    average_yield_per_hectare_per_year = total_yield / (self.params.area_total *
self.params.year_count) if total_yield > 0 else 0
    average_cost_per_hectare_per_year = total_cost / (self.params.area_total *
self.params.year_count) if total_yield > 0 else 0
    average_cost_per_kg_per_year = total_cost / (total_yield * 1000) if total_yield > 0
else 0
    average_profit_per_hectare_per_year = total_profit / (self.params.area_total *
self.params.year_count) if total_yield > 0 else 0
    average_profit_per_kg_per_year = total_profit / (total_yield * 1000) if total_yield >
0 else 0

    # Return results in a dictionary
    print("Total profit:", total_profit)
    print("Total cost:", total_cost)
    print("Total yield (tons):", total_yield)
    print("Average yield per hectare per year (tons/ha/year):",
average_yield_per_hectare_per_year)
    print("Average cost per hectare per year:", average_cost_per_hectare_per_year)
    print("Average cost per kilogram per year:", average_cost_per_kg_per_year)
    print("Average profit per hectare per year:", average_profit_per_hectare_per_year)
    print("Average profit per kilogram per year:", average_profit_per_kg_per_year)
```

- problems.py

```
from constants import *
import pandas as pd
import numpy as np
from scipy.stats import norm
import random
import time
from sp import (
  SpModel,
  SpParams,
)

print("Chance constraint")
print(f"Number of years {year_count}")
print(f"Number of iterations {iterations}")
print(f"desired_reliability_trader {desired_reliability_trader}")
print(f"percentile {percentile}")
```

```
print(f"number of runs {num_runs}")

results = []

for _ in range(num_runs):
    # Read 100 rows data for 3 crops
    df_crop_prices_history = pd.read_csv("../data/crop_prices_history.csv") * 1000

    percentile_prices = np.percentile(df_crop_prices_history, percentile, axis=0)

    prices_byproduct = []
    prices_trader = []
    prices_market = []

    for year in range(year_count + 1):
        year_data_trader = []
        year_data_market = []
        year_data_byproduct = []

        for crop in range(crop_count):
            crop_data_trader = \
                [[percentile_prices[crop] for scenario in range(iterations)] for _ in
                 range(trader_count)]
            crop_data_market = \
                [[percentile_prices[crop] for scenario in range(iterations)] for _ in
                 range(market_count)]
            crop_data_byproduct = \
                [percentile_prices[crop] * 0.3 for scenario in range(iterations)]

            year_data_trader.append(crop_data_trader)
            year_data_market.append(crop_data_market)
            year_data_byproduct.append(crop_data_byproduct)

        prices_trader.append(year_data_trader)
        prices_market.append(year_data_market)
        prices_byproduct.append(year_data_byproduct)

    # Compute expected profit using SPModel
    sp_params = SpParams(
        area_total=area_total,
        costs_water=costs_water,
        costs_light=costs_light,
        costs_harvest=costs_harvest,
        costs_cut=costs_cut,
        costs_process=costs_process,
        costs_worker_fixed=costs_worker_fixed,
```

```
        costs_worker_hired=costs_worker_hired,
        amounts_water_required=amounts_water_required,
        amounts_light_required=amounts_light_required,
        areas_initial=areas_initial,
        costs_plant=costs_plant,
        amounts_demand_byproduct=amounts_demand_byproduct,
        productivities=productivities,
        amounts_demand_trader=amounts_demand_trader,
        amounts_demand_market=amounts_demand_market,
        prices_byproduct=prices_byproduct,
        prices_trader=prices_trader,
        prices_market=prices_market,
        iterations=iterations,
        count_worker_plant=count_worker_plant,
        count_worker_harvest=count_worker_harvest,
        count_worker_cut=count_worker_cut,
        desired_reliability_trader=desired_reliability_trader,
        desired_reliability_market=desired_reliability_market,
        penalties_demand_trader=[],
        penalties_demand_market=[]
    )
    sp_model = SpModel(sp_params)
    # print("Begin to solve")

    # Record the start time
    start_time = time.time()
    sp_model.solve()
    # Record the end time
    end_time = time.time()

    # Calculate the elapsed time
    elapsed_time = end_time - start_time

    # Print the elapsed time
    print(f"Elapsed Time: {elapsed_time} seconds")

    c = sp_model.problem.constraints
    print(f"Constraints {len(c)}")
    v = sp_model.problem.variables()
    print(f"Variables {len(v)}")

    profit = sp_model.problem.objective.value()
    # print(f"Profit: {profit}")
    results.append(str(profit))

# print(f"Profits = {','.join(results)}")
```

```
# Calculate and display results
results = sp_model.calculate_results()
print(results)
```

- constants.py

```
STDEV_SCALE_FACTORS = [-1.5, 0, 1.5]
iterations = 1
year_count = 8
crop_count = 3
trader_count = 5
market_count = 5
desired_reliability_trader = 0.9
desired_reliability_market = 0.9
num_runs = 1
percentile = 50

area_total = 20
costs_water = 0.1
costs_light = 0.15
costs_harvest = [1700] * (year_count+1)
costs_plant = [[5000] * crop_count] * (year_count+1)
costs_cut = [3000] * (year_count+1)
costs_process = [300] * (year_count+1)
costs_worker_fixed = [5000] * (year_count+1)
costs_worker_hired = [1700] * (year_count+1)
amounts_water_required = [1510, 2470, 3850, 5170, 5170, 5170, 5170, 5170, 5170, 5170]
amounts_light_required = [25000] * crop_count
areas_initial = [
    [0, 0, 0, 0, 20, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 20, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
]
productivities = [
    [
        [0, 50, 60, 80, 90, 90, 90, 80, 50, 30],
        [0, 40, 50, 70, 80, 80, 80, 70, 40, 20],
        [0, 10, 20, 30, 30, 30, 30, 20, 10, 10],
    ]
] * (year_count+1)
amounts_demand_byproduct = [[20, 0, 0]] * (year_count+1)
amounts_demand_trader = [
    [[200] * trader_count, [160] * trader_count, [40] * trader_count]
] * (year_count+1)
amounts_demand_market = [
    [[200] * market_count, [160] * market_count, [40] * market_count]
```

```
] * (year_count+1)

count_worker_plant = [10,0,0,0,0,0,0,10,0,0]

count_worker_harvest = [0,3,3,3,3,3,3,3,3,0]

count_worker_cut = [0,0,0,0,0,0,0,6,0,6]
```