A TWO STAGE STOCHASTIC AND RISK AVERSE APPROACH TO INVENTORY POSITIONING IN FLORICULTURAL AUCTION NETWORK

A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF MIDDLE EAST TECHNICAL UNIVERSITY

BY GİZEM SEZEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN INDUSTRIAL ENGINEERING

SEPTEMBER 2014

Approval of the thesis:

A TWO STAGE STOCHASTIC AND RISK AVERSE APPROACH TO INVENTORY POSITIONING IN FLORICULTURAL AUCTION NETWORK

submitted by **GİZEM SEZEN** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen Dean, Graduate School of Natural and Applied Sciences	
Prof. Dr. Murat Köksalan Head of Department, Industrial Engineering	
Assist. Prof. Dr. Cem İyigün Supervisor, Industrial Engineering Department, METU	
Examining Committee Members:	
Assoc. Prof. Dr. Zeynep Pelin Bayındır Industrial Engineering Department, METU	
Assist. Prof. Dr. Cem İyigün Industrial Engineering Department, METU	
Assoc. Prof. Dr. Sinan Gürel Industrial Engineering Department, METU	
Assoc. Prof. Dr. İsmail Serdar Bakal Industrial Engineering Department, METU	
Assist. Prof. Dr. Özlem Çavuş Industrial Engineering Department, Bilkent University	

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: GİZEM SEZEN

Signature :

ABSTRACT

A TWO STAGE STOCHASTIC AND RISK AVERSE APPROACH TO INVENTORY POSITIONING IN FLORICULTURAL AUCTION NETWORK

SEZEN, GİZEM M.S., Department of Industrial Engineering Supervisor : Assist. Prof. Dr. Cem İyigün

September 2014, 76 pages

This study is motivated by a real life problem of an auction company serving in floricultural sector. Company has 6 auction centers. A supplier chooses the auction center that his or her products will be auctioned and the products are delivered to that auction center before the clock. After the auction, sold products are delivered to the buyer's boxes hired at the auction centers. Some inefficient logistic flows are observed in the network of the company due to the positioning of products before the clock. This study aims to find the initial optimal product positioning decisions prior to the auction that minimizes the transportation cost in the network under the uncertainties of buyers and their purchase quantities. Two-stage stochastic integer programming is proposed to model the problem. A scenario based approach is used in the model. In order to solve large sized problems, L-shaped and multicut L-shaped decomposition algorithms are adapted to the problem. In addition, problem is also modeled with risk-averse approach. First-order mean-semideviation is used as the risk measure. Risk-averse model aims to minimize the expected cost of transportation and expected upper deviation from the expected cost. The objective function of the risk-averse model is nonlinear and nondifferentiable. Linearization method and a variation of multicut L-shaped decomposition algorithm for the linearized formulation are proposed for the solution of risk-averse model.

Keywords: Two-Stage Stochastic Integer Programming, Risk Aversion, L-shaped Decomposition, Multicut L-shaped decomposition, Mean-semideviation, Floriculture, Auction

ÇİÇEKÇİLİK SEKTÖRÜ MÜZAYEDE AĞINDA ENVANTER YERLEŞTİRME İÇİN İKİ AŞAMALI RASSAL VE RİSKTEN KAÇINAN YAKLAŞIM

SEZEN, GİZEM Yüksek Lisans, Endüstri Mühendisliği Bölümü Tez Yöneticisi : Yrd. Doç. Dr. Cem İyigün

Eylül 2014, 76 sayfa

Bu çalışma çiçekçilik sektörü müzayede ağında çalışan bir sirketin gerçek bir probleminden motive olmuştur. Sirketin 6 adet müzayede merkezi vardır. Bir tedarikçi ürününün müzayede görmesini istediği müzayede merkezini kendisi seçer ve müzayede öncesi ürünler o müzayede merkezine teslim edilir. Müzayede sonrasında, satılan ürünler alıcıların müzayede merkezlerinde kiraladıkları alanlara teslim edilir. Ürünlerin, müzayede öncesi müzayede mekezlerine yerleştirim sekli kaynaklı sirket ağında verimsiz lojistik akışlar gözlenmiştir. Bu çalışmanın amacı alıcıların kim olduğu ve satın aldıkları miktardaki belirsizliği dikkate alarak, müzayede a ğında beklenen ulasım maliyetini en azlayan müzayede öncesi optimum ürün yerleştirme kararlarını bulmaktır. Problemi modellemek için iki aşamalı rassal tamsayılı programlama önerilmektedir. Modelde senaryo bazlı bir yaklaşım izlenmiştir. Büyük ölçekli problemleri çözmek için L-biçimli ayrıştırma, çok kesimli L-biçimli ayrıştırma algoritmaları probleme uyarlanmaktadır. Bunların yanısıra, problem riskten kaçınan bir yaklaşımla da modellenmektedir. Risk ölçütü olarak birinci derece ortalama-yarısapma kullanılmıştır. Riskten kaçınan model, beklenen ulaşım maliyetini ve beklenen maliyetten beklenen üst sapmayı en azlamaktadır. Riskten kaçınan modelin amaç fonksiyonu doğrusal olmayan ve türevlenemeyen bir fonksiyondur. Bu fonksiyonu doğrusal hale getirmek için bir metod ve çok kesimli L-biçimli ayrı stırma algoritmasına benzer bir algoritma önerilmektedir.

Anahtar Kelimeler: İki Aşamalı Stokastik Tamsayılı Programlama, Riskten kaçınma, L-biçimli ayrıştırma, Çok kesimli L-biçimli ayrıştırma, Ortalama-yarı sapma, Çiçekçilik ,Müzayede To my family

ACKNOWLEDGMENTS

This thesis has been completed with support of many people. First, I would like to thank my supervisor Cem İyigün for his supports, his understanding and motivating attitude throughout the thesis. I would also like to thank Özlem Çavuş for her valuable contributions, her time, and comments on my thesis.

I sincerely thank to examining committee members Zeynep Pelin Bayındır, Sinan Gürel and İsmail Serdar Bakal for giving their time to review the thesis and offering valuable comments.

I had a chance to work at FloraHolland as a part of this study. I would like to thank Edwin Wenink and Mark Breugem for giving me this great chance and sharing their experience and knowledge with me. I would also like to thank Tom Van Woensel from Eindhoven University of Technology for his support and guidance for this study during my internship at FloraHolland.

I am grateful to Eindhoven University of Technology and the Scientific and Technological Research Council of Turkey (TUBITAK) for their financial support.

I also want to thank my friends Murat Yılmaz, Burcu Kılıç, Gökhan Yılmaz, İrem Şenyurt for their friendship and continuous support for the entire time since we met. You made my life in Ankara amazing. Moreover, my special thanks go to Derya Kılınç, Çiğdem Renkli and Cem Randa for being there whenever I need during my bachelor and master studies. They always listened my concerns, even the unreasonable ones, and calmed me down. Furthermore, I would like to thank Sine Taymaz and Esra Erişen for the great time we spent in Eindhoven. I would also like to thank my dear colleagues for easing my job and enduring me during the most stressful month of my thesis.

Last but not least, I would like to express my deepest gratitude to my parents, and my brother for their love and support throughout my life.

TABLE OF CONTENTS

ABSTR	ACT			v
ÖZ				vii
ACKNO	WLEDO	GMENTS .		x
TABLE	OF CON	TENTS .		xi
LIST O	F TABLE	ES		xiv
LIST OI	F FIGUR	ES		xvi
CHAPT	ERS			
1	INTRO	DUCTION	۱	1
2	LITER	ATURE RI	EVIEW	5
	2.1	Two-stag	e Stochastic Linear Programming with Recourse	5
		2.1.1	Two-stage Stochastic (Mixed) Integer Programming with Recourse	7
	2.2	Risk-aver	rse Optimization	12
		2.2.1	Mean-Semideviation in Two-stage Stochastic Pro- gramming	13
3	CASE I	PROBLEM	AND MOTIVATION	17
	3.1	FloraHol	land	17
	3.2	Problem	Environment	19
		3.2.1	Logistic Flows in FloraHolland Network	19

		3.2.2	Auction Process	21
	3.3	Problem	Definition	22
4	THE RI	SK-NEUT	TRAL MODEL	27
	4.1	Risk-neut	tral Model	27
	4.2	Solution .	Approaches for the Risk-neutral Model	32
		4.2.1	L-shaped Method	32
		4.2.2	Multicut L-shaped Method	34
	4.3	Applicati	on of Solution Approaches to P1	36
		4.3.1	Application of L-shaped Method to P1	37
		4.3.2	Application of Multicut L-shaped Method to P1	39
5	THE RI	SK-AVER	SE MODEL	41
	5.1	The Risk-	-averse Model	41
	5.2	Solution .	Approaches for Risk-averse Model	44
	5.3	Applicati	on of the Solution Approaches to MSD	47
6	COMPU	JTATION	AL STUDY	53
	6.1	Computa	tional Design	53
	6.2	Results of	f the Risk-neutral Model	55
	6.3	Results of	f the Risk-averse Model	60
	6.4	Comparis	son of the Risk-neutral and the Risk-averse Model .	65
7	CONCI	LUSION .		69
REFERI	ENCES			71

APPENDICES

А	COMPARISON OF THE RISK-NEUTRAL AND THE RISK-AVERSE	
	MODEL WITH 2C AND 3C SECOND STAGE UNIT TRANSPORTA-	
	TION COSTS	75

LIST OF TABLES

TABLES

Table 6.1 CPLEX results of P1 with a supply quantity of 100	56
Table 6.2 CPLEX results of P1 with a supply quantity of 1,000	56
Table 6.3 CPLEX results of the LP relaxation of P1 with a supply quantity of100	57
Table 6.4 CPLEX results of the LP relaxation of P1 with a supply quantity of1,000	57
Table 6.5 L-shaped and multicut L-shaped results of the LP relaxation of P1 with a supply quantity of 100	58
Table 6.6 L-shaped and multicut L-shaped results of the LP relaxation of P1 with a supply quantity of 1,000	58
Table 6.7 Value of stochastic solution for a supply quantity 100	60
Table 6.8 Value of stochastic solution for a supply quantity 1000	60
Table 6.9Comparison of CPLEX solver and multicut algorithm for risk-aversemodel with a supply quantity of 100	61
Table 6.10 Comparison of CPLEX solver and multicut algorithm for risk-aversemodel with a supply quantity of 1,000	62
Table 6.11 Comparison of the risk-neutral and the risk-averse models with a supply quantity 100	66

Table 6.12 Comparison of the risk-neutral and the risk-averse models with a supply quantity 1,000	66
Table 6.13 Comparison of the risk-neutral and the risk-averse models with dif-ferent second stage unit costs for a supply quantity 100	67
Table 6.14 Comparison of the risk-neutral and the risk-averse models with dif-ferent second stage unit costs for a supply quantity 1,000	68
Table A.1 Comparison of the risk-neutral and the risk-averse models with $2c$ second stage unit transportation costs for a supply quantity of 100	75
Table A.2 Comparison of the risk-neutral and the risk-averse models with $3c$ second stage unit transportation costs for a supply quantity of $100 \dots \dots$	75
Table A.3 Comparison of the risk-neutral and the risk-averse models with $2c$ second stage unit transportation costs for a supply quantity of 1,000	75
Table A.4 Comparison of the risk-neutral and the risk-averse models with $3c$ second stage unit transportation costs for a supply quantity of 1,000	76

LIST OF FIGURES

FIGURES

Figure 3.1	Overview of Floricultural Supply Chain adapted to FloraHolland	
Netwo	rk [45]	19
Figure 3.2	Current Case: Grower's Inventory Positioning [44]	23
Figure 6.1	Comparison of elapsed times of L-shaped and multicut L-shaped	
algorit	hms	59
Figure 6.2	CPLEX elapsed time for MSDL	63
Figure 6.3	Number of CPLEX iterations for MSDL	64
Figure 6.4	Comparison of elapsed times of CPLEX and multicut algorithm	
for MS	SDL	65

CHAPTER 1

INTRODUCTION

An auction is a method of selling goods, property or services to the highest bidder. The origin of the word auction is the Latin word augeø. It means "I increase" or "I augment" [26]. When you think of auction, mostly art auctions or antique auctions come to mind. However, although it is not known widely, buying and selling cut flowers and plants by auction are also common. Different than the classical auctions, these auctions are descending price auctions. First the auctioneer sets a high price. Then the price is decreased progressively until a bidder agrees on the price. The bidder tells the auctioneer the amount of product he/she wants to buy. If there are left products after a bid, the price continues to be decreased for the next bid. Auction finishes either when all products are sold or when lowest acceptable price of the auctioneer is reached. This type of auction is called Dutch auction or Clock auction [43].

This study is motivated by a real life problem of an auction company serving in floricultural industry in the Netherlands. The company matches growers and buyers in the floricultural industry. Half of the company's sales is made by auctions. There are six auction centers in the company network. Grower chooses the auction center where products will be auctioned. Products need to be physically positioned at that auction center. Buyers can physically present at the auction center or they can remotely buy products from the internet. They have buyer's boxes hired or owned at the auction centers for consolidation, distribution and logistic purposes. Products are sold by a Dutch auction. After the auction, sold products are delivered to the corresponding buyers' boxes located at auction centers.

Dat [45] states that 70% of the products positioned to an auction center from the hin-

terland of that auction center while 30% comes from the hinterland of some other auction center. About 7% out of that 30% are transported back to the auction center closest to the area where the products grew and came from before the auction. More-over, some products are transported between more than one auction center which results in more transportation and handling costs and loss of time. These observations are the foundation of our case problem.

In this study, we relax the need of products' presence at the auction center for the auction. However, products still need to present in the company's auction center network before the auction because they need to be checked for type, quantity, length, volume and quality.

Our aim is to optimally position the products at auction centers prior to the clock minimizing the cost and considering the uncertainties about who the buyers of the product will be and what their purchase quantities will be.

Stochastic programming is used to model optimization problems that incorporates uncertainty, which are represented as random variables. It is widely used in many application areas such as finance, production planning, energy planning, healthcare management, water resource management due to uncertainties involved in real life problems [1]. Two-stage stochastic programming is a class of stochastic programming, in which a set of decisions are taken at each stage. First-stage decisions are subject to uncertainty while second-stage recourse actions are decided when the uncertainty is resolved. Two-stage stochastic problem formulation is a suitable choice to model our case problem because uncertainties involved in the problem are resolved at the auction. Before the auction, we are not fully informed about the buyers of the product and their purchase quantities. Buyer and the purchase quantities of each buyer are revealed during the auction; thus, we are informed about the number of products that will be repositioned in the auction center network. Then, products are relocated in the auction center network accordingly. In our problem environment, initial product positioning before the clock is the first-stage decision, and repositioning process after the clock is the second-stage decision.

Moreover, in stochastic programming, objective function is generally to minimize (maximize) expected cost (profit). Therefore, the expected value models implicitly assume that the decision maker is risk-neutral. However, using expected value may not be the best way to model some optimization problems because of the risk involved in the processes. Lately risk-averse preferences in stochastic programming has drawn more attention. Utility functions, stochastic dominance constraints, valueat-risk, mean-risk measures and coherent risk measures are several options to involve risk-aversion into the problem context.

This study is composed of two main parts. In the first part, we use two-stage stochastic integer programming to formulate the case problem in a risk-neutral environment. In the second part we introduce risk-aversion into the problem. We use first-order mean-semideviation risk measure, which is a coherent risk measure, in the objective function and again model the case problem as two-stage stochastic integer program.

L-shaped method and its variations are the most frequently used methods to solve two-stage stochastic programs [1]. For larger problems we propose L-shaped method and Multicut L-shaped method for the risk-neutral case and a variation of Multicut L-shaped method for risk-averse case. Furthermore, we compare and comment on the computational efficiencies of the methods.

The organization of our study is as follows. In Chapter 2, literature review about twostage stochastic programming and risk-averse optimization is presented. As indicated before, this study is motivated by a real life problem of a company, namely FloraHolland. In Chapter 3, FloraHolland and the case problem are introduced. In Chapter 4 risk-neutral environment is assumed to model and solve the case problem. Chapter 5 is devoted to the risk-averse approach for the case problem. Chapter 6 provides the numerical study about the models and the solution approaches. Thesis is concluded in Chapter 7.

CHAPTER 2

LITERATURE REVIEW

Literature review is introduced in two main parts. First, two-stage stochastic programming is described in Section 2.1. Second, risk-averse optimization is explained in Section 2.2. First part starts with introducing general information and the formulation of two-stage stochastic programming. Then we focus on two-stage stochastic integer programming formulation and the solution methods used in the literature. In the second part, risk-averse optimization is introduced. Mean-semideviation risk measure and its applications in two-stage stochastic programming are described.

2.1 Two-stage Stochastic Linear Programming with Recourse

In this section, general formulation of two-stage stochastic programming with recourse are given. Formulations and discussions are based on Birge and Louveaux [1].

Two-stage stochastic programs with recourse are originated by Dantzig [2] and Beale [3]. They have two components; one is free of uncertainty and the other one is subject to uncertainty. Uncertainty means that some of the problem data can be represented as random variables. There are two sets of decisions. A set of decisions is taken without full information on the future outcome of the uncertainty. Although full information is not available, it is assumed that a probabilistic description of the random variables is available such as probability distribution, density. This set of decisions are called first-stage decisions. They are usually represented by a vector x. Later, when the uncertainty is eliminated (i.e. full information is received on the realization of an event),

second-stage decisions (i.e. recourse actions) are taken. These decisions are usually represented by a vector y, $y(\omega)$ or $y(\omega, x)$ if second-stage decision function differs depending on both the outcome of the random experiment and first-stage decision.

The classical two-stage stochastic programming formulation is as follows:

Minimize
$$z = c^T x + \mathbb{E}_{\xi}[\min q(\omega)^T y(\omega)]$$
 (2.1)

subject to
$$Ax = b$$
, (2.2)

$$T(\omega)x + Wy(\omega) = h(\omega), \qquad (2.3)$$

$$x \ge 0, y \ge 0 \tag{2.4}$$

where $x \in \mathbb{R}^{n_1}$ is first-stage decision vector, $c \in \mathbb{R}^{n_1}$, $b \in \mathbb{R}^{m_1}$ and $A \in \mathbb{R}^{m_1 \times n_1}$ are the first-stage vectors and matrices corresponding to $x, \omega \in \Omega$ is the realized second-stage random event, $q(\omega) \in \mathbb{R}^{n_2}$, $h(\omega) \in \mathbb{R}^{m_2}$ and $T(\omega) \in \mathbb{R}^{m_2 \times n_1}$ are the second-stage problem data for the realization ω . ω influences all components of ξ where $\xi^T(\omega) = (q(\omega)^T, h(\omega)^T, T_1(\omega), ..., T_{m_2}(\omega))$. $\mathfrak{E} \subset \mathbb{R}^N$ is defined as the support of ξ , the smallest subset in \mathbb{R}^N such that $P(\mathfrak{E}) = 1$. The first part of the objective function $(c^T x)$ is deterministic, and the second part $\mathbb{E}_{\xi}[(q(\omega)^T y(y))]$ is an expectation taken over all realizations of the random event ω . For each ω , $y(\omega)$ is the solution of a linear program; thus *deterministic equivalent program* notation is as follows :

Minimize
$$z = c^T x + \mathfrak{Q}(x)$$

subject to $Ax = b$, (2.5)
 $x \ge 0$

where $\mathfrak{Q}(x) = \mathbb{E}_{\xi}[Q(x,\xi(\omega))]$ is the expected second-stage function, $Q(x,\xi(\omega)) = \min_{y} \{q(\omega)^{T}y | W(\omega)y = h(\omega) - T(\omega)x, y \ge 0\}$ is the second-stage value

 $Q(x,\xi(\omega)) = \min_{y} \{q(\omega)^{T} y | W(\omega)y = h(\omega) - T(\omega)x, y \ge 0\}$ is the second-stage value function.

Stochastic components vector ξ could be continuously distributed or could be a discrete random variable. Our focus in this study is on the case where ξ is a discrete random variable; which is widely preferred in applications. With a finite discrete random variable ξ , $\mathfrak{Q}(x)$ and $Q(x, \xi(\omega))$ can be formulated as follows:

$$\mathfrak{Q}(x) = \mathbb{E}_{\xi}[Q(x,\xi(\omega))] = \sum_{s=1}^{S} p_s Q(x,\xi_s)$$
$$Q(x,\xi_s) = \min_{y_s} \{q_s^T y_s | W_s y_s = h_s - T_s x, y_s \ge 0\}$$

where s = 1, ..., S represents the S realizations of ξ and p_s represents the probability of realization s. Moreover, combining two stages together we can reformulate the deterministic equivalent which is referred as *extensive form* in [1] as follows:

$$\begin{aligned} \text{Minimize} \quad z &= c^T x + \sum_{s=1}^{S} p_s q_s^T y_s \\ \text{subject to} \quad Ax &= b, \\ W_s y_s &= h_s - T_s x \quad \forall s \in S, \\ x &\geq 0, \quad y_s \geq 0 \quad \forall s \in S. \end{aligned}$$

$$(2.6)$$

2.1.1 Two-stage Stochastic (Mixed) Integer Programming with Recourse

Formulation (2.1)-(2.4) is the simplest form of the two-stage stochastic programming. Extensions can be modeled easily. One of the extensions is two-stage stochastic (mixed) integer programming. Formulation (2.1)-(2.4) can be easily modified by changing constraint (2.4) with a general form:

$$x \in \mathbb{R}^{n_1-r_1}_+ \times \mathbb{Z}^{r_1}_+, \quad y(\omega) \in \mathbb{R}^{n_2-r_2}_+ \times \mathbb{Z}^{r_2}_+,$$

where $0 \le p_1 \le n_1$ and $0 \le p_2 \le n_2$.

Several variations are possible for two-stage stochastic (mixed) integer programming. For example, both of the stage variables can be purely integer (i.e. $r_1 = n_1$ and $r_2 = n_2$) or mixed integer (i.e. $0 < r_1 < n_1$ and $0 < r_2 < n_2$). Variables of one of the stages can be continuous (i.e. $r_1 = 0$ or $r_2 = 0$) and the other stage variables can be (mixed) integer. Furthermore, the integer variables can be restricted to be binary variables.

It is known that solving integer problems and solving stochastic problems are difficult. Two-stage stochastic (mixed) integer programming combines the difficulties of both problems. Three levels of difficulties are mentioned by Ahmed [7] for solving twostage stochastic (mixed) integer problems which do not have continuous variables at any stages.

1. Evaluating the second-stage cost for a fixed first-stage decision and a particular realization of the uncertain parameters. This step requires to solve many second-stage problems which can be NP hard integer problems; thus it is computationally difficult.

- 2. Evaluating the expected second-stage cost for a fixed first-stage decision. If the variables have continuous distribution, this step requires integrating the value function of an integer program; which is generally impossible. In the case of discrete distribution huge number of similar integer programs are required to be solved; which makes the problem computationally difficult.
- 3. *Optimizing the expected second-stage cost.* The value function of an integer program is non-convex and often discontinuous, indeed it is lower semicontinuous. Hence, the expected second-stage cost function is non-convex in *x*. This complex objective function causes many difficulties in optimization.

However, if the second-stage variables are continuous (i.e. $r_2 = 0$), much of the theory and algorithms for two-stage stochastic linear programs, which do not rely on convexity of first-stage constraints, can be applied because the problem has a convex objective subject to mixed integer constraints.

Solution algorithms for two-stage stochastic integer programs are based on decomposition methods. Decomposition algorithms can be classified into two; stage-wise (primal) decomposition algorithms and scenario-wise (dual) decomposition algorithms. Most of the solution algorithms use stage-wise decomposition. These algorithms are fundamentally variations of Benders' decomposition, L-shaped method. They mainly differ in how second-stage value function is approximated and updated. In scenario wise decomposition, the idea is to introduce a copy of first-stage variable x for each scenario $s \in \{1, ..., S\}$. Then, nonanticipativity constraint ($x_1 = x_2 = ... = x_S$) is added to the model to provide that the first-stage decision should not depend on the scenario which will prevail in the second-stage. Afterwards Lagrangian relaxation of the problem is taken with respect to nonanticipativity condition. In Lagrangian dual, the problem is separated into subproblems for each scenario. Lagrangian dual is a convex non-smooth program which can be solved by subgradient methods [8].

Besides the decomposition method used, studies on the solution methods of two-stage stochastic integer programs can also differ in the integrality restrictions on the vari-

ables in the stages. Wollmer [9] and Laporte and Louveaux [10] study two-stage mixed integer programming with binary first-stage variables and continuous secondstage variables. In [9], Wollmer applies a variation of Benders' decomposition by incorporating an implicit enumeration scheme which is fundamentally a backtracking procedure for methodically searching through the possible solutions. Laporte and Louveaux [10] introduce integer L-shaped method. They study a minimization problem. The method is a combination of Branch and Bound algorithm and the classical L-shaped method. Nodes are created by Branch and Bound method and for each node L-shaped algorithm is applied to solve the current two-stage problem. If integrality restriction is violated at a node two new branches are created. At a node if the current problem does not have any feasible solution or the optimal value of the node is higher than the current best objective function value, the node is fathomed. Different than the classical Branch and Bound algorithm, nodes are not necessarily fathomed when integrality conditions are satisfied, a new optimality cut is obtained from that solution and the problem is solved again.

Carøe and Schultz [8] work with mixed integer variables at both stages in a maximization problem. They use a scenario wise decomposition technique combined with a Branch and Bound method. In Branch and Bound, a node is selected and started to be solved by Lagrangian dual, optimal value of which provides an upper bound for the problem. If the upper bound is higher than the current best solution, node is fathomed. If the first-stage x_s values are not identical, the average is taken and by some rounding heuristics, it is rounded to obtain feasibility. A new objective value is obtained using the integer x values and objective value is updated. At the branching part, the component x_i is used to generate two new branches by adding some constraint for x_i . Zheng et al. [11] study stochastic mixed integer program in both stages to solve a unit commitment problem, which is an important optimization problem in power system operations and control. Their solution method is based on Bender's decomposition and integer L-shaped method.

Escudero et al. [12] study two-stage stochastic mixed 0-1 problem. There are two types of variables in both first and second stages. One type of the variables are continuous and the others are binary. They introduce a variation of Lagrangian decomposition, namely Cluster Based Lagrangian Decomposition for obtaining strong lower bounds to solve the problem. They propose to decompose the model into a set of scenario clusters where nonanticipativity constraints are implicitly considered.

Ntaimo [13] studies with binary first-stage and mixed integer second-stage variables. The author introduce a new cutting plane method; Fenchel decomposition (FD). FD cuts, which are used in the decomposition algorithm, are based on the Fenchel cuts from integer programming. Fenchel decomposition is also a modified L-shaped algorithm. Second-stage variables are relaxed and solved by L-shaped method, then FD cuts are generated and added to the subproblem and subproblem is solved again. When the solution of the subproblem is integer, the optimality cuts which are obtained from dual multipliers are added to the master problem. Gade et al. [14] study with binary first-stage and integer second-stage variables. They propose a method based on Bender's decomposition which utilizes Gomory cuts.

Carøe and Tind [15] and Ahmed et al. [16] study models with mixed integer first-stage and integer second-stage variables. In [15], integer recourse is studied. Second-stage cost vector is assumed to be fixed. The proposed method is based on L-shaped method and duality theory. Dual price functions are used to generate feasibility and optimality cuts. It is shown that finite convergence is achieved if second-stage problem is solved by Branch and Bound algorithm or Gomory's fractional cutting plane algorithm. In [16], a finite Branch and Bound algorithm, which avoids explicit enumeration of all discontinuous pieces of the value function, is proposed. They assume that technology matrix is fixed. The idea behind their work is to reformulate the problem by variable transformation which triggers a special structure to the discontinuities of the value function. At branching, the discontinuities are eliminated. In the absence of discontinuities, exact representation of the value function is provided at bounding.

Hemmecke and Schultz [17] propose a different decomposition method which is neither stage-wise decomposition nor scenario-wise decomposition. Instead of decomposing the problem itself, they decompose test sets. Test sets are finite collections of vectors enabling the solution of integer linear programs by simple augmentation procedures. They use Graver test sets to solve the problem. Variables in both stages are integers and only right hand side vector is uncertain, the rest are assumed to be fixed. They propose to decompose the test sets and use the decomposed test sets to solve the problem by a simple augmentation process. Kong et al. [18] also study with integer variables in both stages. In the study, only randomness is in the right hand side as in [17]. Coefficient matrix is assumed to be integral. They reformulate the two-stage stochastic integer problem by an equivalent superadditive dual formulation that uses the value functions in both stages. In order to find the value functions, they propose two different algorithms; an integer programming based algorithm and a dynamic programming based algorithm. Then, they develop a global Branch and Bound and level-set approach to solve the new formulation.

Sherali and Fraticelli [19], Sen and Higle [20], Sen and Sherali [21] study two-stage stochastic integer problems with binary first-stage variables and 0-1 mixed integer second-stage variables. In [19], a modified Benders' decomposition is proposed. Subproblems are solved by Reformulation Linearization Technique and lift and project cutting plane scheme and cuts are generated as functions of first-stage variables. Thus, cutting planes of a scenario can be reused in the same scenario in the next Bender's iteration by only updating the values of first-stage variables. In [20], a stage-wise decomposition algorithm is used. Recourse matrix and second-stage cost vector are assumed to be fixed. They characterize the convexifications (relaxations) of the secondstage problem by disjunctive programming theory. One of the important observations of the authors is that second-stage convexifications associated with different scenarios have common cut coefficients for y when the recourse matrix is fixed. This means that a valid inequality for a scenario is easily used to derive the valid inequality for the other scenario; which is referred as Common Cut Coefficients Theorem. The proposed algorithm, namely Disjunctive Decomposition algorithm, benefits from this theorem. They work with master and subproblems that are convexifications of two coupled disjunctive programs. In [21], Sen and Sherali continue dealing with convexification of stochastic mixed integer programming in the line of work initiated by [19] and [20]. They incorporate Branch and Cut methods for the second-stage problem in a disjunctive decomposition algorithm. Second-stage problems are solved by Branch and Cut method and cuts added to the master problem are generated by disjunctive approach. In the algorithm, they benefit from the fact that a large stochastic mixed integer program can be solved by dividing it into small mixed integer programs which can be solved in parallel. Disjunctive programming is used also in [39] and [40]. However, different than the previously introduced papers, first-stage variables are continuous. Second-stage variables are again 0-1 mixed integers. In both studies lift and project cuts are used in disjunctive decomposition algorithm.

2.2 Risk-averse Optimization

Stochastic optimization models optimize the expected random outcome in risk-neutral models. However, if someone is interested in the fluctuations of the realizations of the outcomes and the risk involved due to the fluctuations, risk-averse optimization is a better way to model stochastic optimization problems [22]. In the literature, there are different approaches to model the risk-aversion in optimization problems since the risk perception may change depending on the decision maker [6]. A classical methodology is using expected utility theory to model risk-aversion. Objective function is formulated as follows:

$Min_{x \in X} \mathbb{E}[u(F(x, \omega))]$

where $F : \mathbb{R}^n \times \Omega \mapsto \mathbb{R}$ is the cost function, $X \subset \mathbb{R}^n$ is the feasible set and *u* is the disutility function, which is nondecreasing and convex for minimization problems.

The problematic part of using expected utility theory is to define utility functions. Moreover, if some arbitrary functions are included, the solutions may not result in meaningful interpretations [22].

Mean-risk models are another way of incorporating risk-averse preferences into the optimization models. They combine the mean, which is the expected outcome, and risk, which is a scalar measure of variability of the outcomes. Mean-risk objective function is defined as

$$Min_{x\in X}\mathbb{E}[Z_x]+\kappa\mathbb{D}[Z_x],$$

where Z_x is the random outcome, a non-negative scalar κ is the trade-off coefficient and \mathbb{D} is the dispersion function to measure the risk associated with Z_x . There are different risk measures used in mean-risk models such as variance, semivariance, value-at-risk (VaR), conditional value-at-risk (CVaR), central deviation and semideviation. Mean-variance model is introduced by Markowitz [23] for the context of portfolio management. Variance and semivariance models have been used in many studies since then. VaR, CVaR, central deviation and semi-deviation models are preferred by many researches in the recent studies.

As their names suggest, VaR and CVar models are related to each other. Let Z be a random variable representing losses, then VaR and CVar is defined as follows [22]:

$$VaR_{\alpha}[Z] = \inf\{t : P(Z \le t) \ge 1 - \alpha\} = \inf\{t : P(Z > t) \le \alpha\}$$

$$CVaR_{\alpha}[Z] := \inf_{t \in \mathbb{R}} \{t + \alpha^{-1}\mathbb{E}[Z - t]_{+}\}$$

Researchers prefer CVaR to VaR. This is because VaR is not a coherent risk measure (see Artzner et al. [24]). On the other hand, CVaR is a coherent risk measure, which is advantageous in optimization problems. CVaR is studied in the context of portfolio management in [27] and [28], in the context of waste management in [29], in the context of hydrocarbon biorefinery supply chain in [30], in the context of disaster management in [31], in the context of reverse logistics network design in [32], in the context of water resources allocation in [33], in the context of natural gas transmission network expansion and LNG terminal location planning in [34].

In this study, we use mean-semideviation as the risk measure. The next section focuses on mean-semideviation model and its application in two-stage stochastic programming.

2.2.1 Mean-Semideviation in Two-stage Stochastic Programming

Upper (lower) semideviation measures are appropriate for minimization (maximization) problems. In this study, we work with a cost minimization objective function. Thus, upper semideviation is used in this thesis. Upper semideviation of order p is defined as:

$$\delta_p[Z] = (\mathbb{E}[(Z - \mathbb{E}[Z])_+^p])^{\frac{1}{p}}$$

where $Z : \Omega \mapsto \mathbb{R}$ is a random variable which represents cost and belongs to the space $X_p = \mathcal{L}_p(\Omega, F, P)$ for $p \ge 1$. Then, the corresponding mean-semideviation model takes the general form in [22];

$$Min_{x \in X} \quad \mathbb{E}[Z_x] + \kappa \delta_p[Z_x] \tag{2.7}$$

In the context of stochastic programming problem of the form

$$\operatorname{Min} \{\mathbb{E}[f(x,\omega)] : x \in X\},\$$

where $f(x, \omega) = c^T x + Q(x, \xi(\omega)); Q(x, \xi(\omega))$ is the second-stage value function for a given realization $\omega; x \in \mathbb{R}^n$ is a vector of decision variables; $X \subset \mathbb{R}^n$ is a nonempty set of feasible decisions; (Ω, F, P) is a probability space with elements ω ; and $f : \mathbb{R}^n \times \Omega \mapsto \mathbb{R}$ is a cost function such that $f(., \omega)$ is convex for all $\omega \in \Omega$, and f(x, .)is *F*-measurable and *P*-integrable for all $x \in \mathbb{R}^n$, the mean-semideviation model is formulated as follows [35]:

$$Min_{x \in X} \quad \mathbb{E}[f(x,\omega)] + (\mathbb{E}[(f(x,\omega) - \mathbb{E}[f(x,\omega)])_+^p])^{\frac{1}{p}}.$$

The aim of the model is to penalize the excess of $f(x, \omega)$ over its mean [22].

Semideviation is a coherent risk measure; which provides advantages in optimization. Coherent risk measure term is introduced in [24]. Risk measure is a function $\rho(Z)$ which maps Z into \mathbb{R} where Z is a real valued random function defined on space \mathcal{Z} . Moreover, a risk measure is called coherent if it satisfies the following axioms [22] :

- Translation equivariance: If $a \in \mathbb{R}$ and $Z \in \mathcal{Z}$, then $\rho(Z + a) = \rho(Z) + a$.
- Convexity: $\rho(tZ + (1 t)Z') \le t\rho(Z) + (1 t)\rho(Z')$ for all $Z, Z' \in \mathbb{Z}$ and all $t \in [0, 1]$.
- Positive homogeneity: If t > 0 and $Z \in \mathbb{Z}$, then $\rho(tZ) = t\rho(Z)$.
- Monotonicity: If $Z, Z' \in \mathbb{Z}$ and $Z \ge Z'$, then $\rho(Z) \ge \rho(Z')$.

Besides being a coherent measure, semideviation is proved to be consistent with second degree stochastic dominance (SSD) relation for absolute semideviation (i.e. p = 1) and standard semideviation (i.e. p = 2) provided that κ is bounded by 1 [36]. It is important that a risk measure is consistent with SSD because if X dominates Y under SSD rules, it means that X is preferred to Y within all risk-averse models [36]. In addition, Ahmed [35] shows that mean-semideviation objective (2.7) is convexity preserving for all $p \ge 1$ and $\kappa \in [0, 1]$; thus, it is appropriate for optimization. Consequently, semideviation is a preferable risk measure for optimization problems.

In the following paragraphs, we focus on the applications of mean-semideviation models in two-stage stochastic programs.

With a finite set Ω of scenarios, Ahmed [35] presents a specific formulation for the deterministic equivalent of mean absolute semideviation model. Deterministic equivalent of mean absolute semideviation model does not have a dual-block angular structure. Hence, classical decomposition methods such as L-shaped method are useless. A cutting plane decomposition algorithm, which is a slight variation of the classical methods, is proposed in [35] in order to solve the model with convexity preserving mean absolute semideviation objective function. Kristoffersen [37] also studies semideviation risk measure in two-stage stochastic linear programming problems. Assumptions of the paper are complete recourse, dual feasibility, that random vector ξ has finite second moment and that ξ is discrete and finite. Working with these assumptions and p = 1, Kristoffersen [37] also presents a deterministic equivalent formulation for mean-semideviation model using the straight forward calculations in [41].

$$\mathbb{E}[f(x,\omega)] + \kappa(\mathbb{E}[\max\{f(x,\omega) - \mathbb{E}[f(x,\omega)], 0\}])$$

=(1 - \kappa) \mathbb{E}[f(x,\omega)] + \kappa \mathbb{E}[\max\{f(x,\omega), \mathbb{E}[f(x,\omega)]\}] (2.8)

and linearizes it in a way that enables using a variation of Multicut L-shaped method. Kristoffersen's algorithm [37] differs from [35] in terms of the number of cuts added in each iteration and the information within the cuts. We also use the decomposition algorithm in [37] to solve our problem. More information about the paper is given in Chapter 5. Märkert and Schultz [41] apply mean-semideviation objective to two-stage stochastic (mixed) integer programs. With the assumption of complete recourse, sufficiently expensive recourse, that random vector ξ has finite first moment and that ξ is discrete and finite, they derive the structural properties of the resulting model. They also formulate a linearized deterministic equivalent mixed-integer model with the objective (2.8), which resembles that of Kristoffersen in [37]. Although, the proposed linearized model can be solved by a general mixed integer linear programming solver, first algorithmic ideas are proposed in [41] for higher number of scenarios. The logic behind the proposed algorithm is to take (2.8) as nonconvex global optimization problem and to handle it with branch and bound methods. For bounding process, applying Lagrangian relaxation on the linearized version of (2.8) is suggested.

Liu et al. [38] use two-stage stochastic mixed integer problem with a mean semideviation objective to model network retrofit problem of allocating limited resources over the highway bridges of a transportation system so that structural and travel delay loss of the transportation system will be minimized in case of an earthquake. Because of the high uncertainty involved and their choice of a robust system design, they use a risk-averse objective for the stochastic optimization problem. First-stage variables are binary retrofit decisions. Second-stage has a multicommodity min-cost network flow formulation. They assume finite discrete ξ . Each realization defines a damage scenario. They present an extension of L-shaped decomposition; which is based on the basic ideas of decomposition, linearization and successive approximation.

Miller and Ruszczyński [25] extend risk-averse two-stage stochastic linear programming model by considering an unresolved uncertainty after the second-stage. They use conditional risk measures in the formulations including conditional mean- semideviation. Moreover, the problem is reformulated using dual representation, which is a key property of conditional risk measures. In order to solve the proposed reformulation, a multicut decomposition algorithm which is similar to L-shaped method is presented.

CHAPTER 3

CASE PROBLEM AND MOTIVATION

In this section, the problem studied in this thesis will be explained. In section 3.1, the motivation behind the problem will be described. In section 3.2, problem environment will be introduced. In section 3.3, the case problem will be defined.

3.1 FloraHolland

Floricultural industry is concerned with production and sales of flowers and plants for outdoor and indoor use. When we look at floricultural industry, we see that there are many parties involved in the floricultural supply chain. These are:

- Seed growers/improvers
- Growers
- Intermediary companies
- Traders/exporters
- Whole sale, retail, supermarkets, chain stores
- Transporters/logistics service providers
- End customer [45]

For a successful supply chain, all parties should work together in collaboration. Intermediary companies have a significant role for the success of the floricultural supply chain because they match the suppliers and the buyers. This research is motivated by a real life problem of an intermediary company which serves in floricultural industry in the Netherlands, the heart of the international floriculture sector. This company is FloraHolland.

FloraHolland is one of the key players serving in floricultural industry. It dominates the floriculture sector with a 98% market share in the Netherlands. The cooperative generated 4.5 billion Euros turnover in 2013 [42].

FloraHolland makes an intermediary service between 7,000 growers and 2,400 buyers either by the direct sales of the firm FloraHolland Connect or by the auction sales. In 2013, 52% of the sales was by the auctions and 48% of them was by the direct sales. In the case of direct sales, buyers and growers are connected with predetermined price, quantity and delivery times by the firm FloraHolland Connect. However, in the case of auction sales, we cannot mention predetermined price, quantity and delivery times. The prices of the products are determined by a Dutch auction during the sales. Dutch auction is also known as clock auction. First, a high price is set by the auctioneer. Then, this price is gradually decreased till a buyer accepts the price or the seller's minimum acceptable price is reached. When the price is accepted, the buyer also specifies the amount of products he/she wants to buy. Therefore, neither the price nor the quantity of the purchased products are predetermined in the case of a clock auction.

FloraHolland is the world's largest auction organization. Every day 38 auction clocks are in operation at FloraHolland auction centers meaning 125,000 auction transactions every day, 12 billion cut flowers and over half a million plants a year. FloraHolland owns 6 auction centers; 5 of which are in the Netherlands. Aalsmeer, Naaldwijk and Rijnsburg auction centers mainly serve the international markets. Bleiswijk and Eelde serve the regional markets. Herongen auction center is in Germany and is a joint venture with Landgard. It also serves the regional market.

FloraHolland is a cooperative business owned by its 5,000 members who are the growers supplying to the company. FloraHolland is originally developed from the idea of collaboration. A century ago, growers came together and started offering their products to the dealers in one place. This made them stronger in the face of

dealers, and they got better prices for their flowers and plants. Since then, FloraHolland has not focused on generating profit. The firm focuses on offering its members the best sales opportunities at the lowest possible cost. The vision and the mission of the company is to maintain and increase its strong position in an upscaling and internationalizing market. FloraHolland aims to tie international and national commerce flows to marketplaces by providing the best and the highest variety. In order to prevent its competitive advantage and its leading position, it needs to innovate and continuously improve its processes.

3.2 **Problem Environment**

In this section, first we briefly introduce all logistic flows in FloraHolland network and then we focus on the auction process and the auction related logistic flows.

3.2.1 Logistic Flows in FloraHolland Network

Due to its role of matching suppliers and buyers, there are many logistic flows involved in FloraHolland network. Dat [45] adapted floricultural supply chain to FloraHolland and illustrated the logistic flows in FloraHolland network as in Figure 3.1.

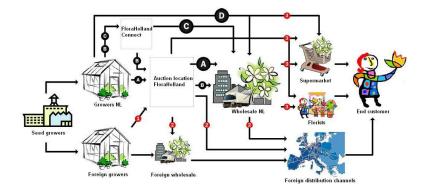


Figure 3.1: Overview of Floricultural Supply Chain adapted to FloraHolland Network [45]

The letters A, B, C, D in Figure 3.1 refer to different types of flows in FloraHolland network. These are clock flows, connect flows, BDO (Buiten de Distributie Om) flows, and BVO (Buiten de Veiling Om) flows.

- A. <u>Clock Flows</u>: These are all flows involved in the auction process. Growers deliver their products to auction centers for the auction. There are two possibilities here. Growers may deliver their products to the auction center where products will be auctioned. Or grower may deliver his/her products to the closest auction center and FloraHolland transports these products to the auction center where they will go through the auction process. Prices of the products are determined during the clock by a Dutch auction. Then, sold products are transported to buyer boxes which can be located at any auction center. Box is a place hired or owned by a buyer in one or several auction locations. It is used for consolidation, distribution, and logistic operations of the buyer.
- B. <u>Connect Flows</u>: These are the flows of the products that are sold through FloraHolland Connect. Suppliers deliver sold products at any auction center with an Electronic Delivery Form (EAB) containing the name and location of the buyer. Then, they are transported to the auction center where buyer's box is located.
- C. <u>BDO (Buiten de Distributie Om) Flows</u>: These flows are also the flows of the products that are sold through FloraHolland Connect. However, different than the connect flows they bypass the FloraHolland distribution network. They are directly delivered to the buyer boxes or their related Logistic Service Providers.
- D. <u>BVO (Buiten de Veiling Om) Flows</u>: These flows are direct flows from growers to buyers bypassing FloraHolland completely. Products are delivered from the grower's location to the buyer's location other than the buyer boxes at the auction centers. During this transportation, products do not pass through any auction center. If a grower is a member of FloraHolland, he/she can sell only a limited amount of product bypassing FloraHolland.

Moreover, there is a distinction between import and export flows in Figure 3.1. Numbers 1, 2, 3 refer to import, export and local Dutch sales respectively. Import products are transported to Netherlands by sea or air freight. They enter FloraHolland Distribution network at Aalsmeer, Naaldwijk or Rijnsburg auction centers. If they will be sold by an auction, FloraHolland transports them to the related auction center before the clock.

In addition, purchased products are delivered to buyers' boxes, the buyers further transport them to their clients. However, these flows are out of the scope of the project because they are outside of the FloraHolland distribution network.

The first two flows; namely Clock Flows and Connect Flows, can be influenced by FloraHolland due to the fact that these flows enter the FloraHolland distribution network. Our focus in this study is on Clock Flows. Connect Flows, BDO flows and BVO flows are out of the scope of this study. Hence, in the next section, we will illustrate the journey of the products during an auction process starting from the grower's location.

3.2.2 Auction Process

There are 6 auction centers in FloraHolland network. In the current situation, growers choose the auction center where their products will go through the clock process. The reasoning behind this choice may vary depending on the grower. Growers can choose the auction center according to its proximity; their belonging to the captive hinterland of the auction center. Moreover, they can choose the auction centers where many buyers are located by considering the chance of getting higher prices at these auction centers. In addition, tradition can also play a role for positioning the products.

After deciding on the auction center where they want their products to be auctioned, they either transport their products directly to that auction center or deliver them to the closest auction center and FloraHolland makes sure that products are delivered to the auction center, where they will be auctioned, before the clock. If a product will be auctioned at auction center A, it needs to be physically located at auction center A in the current system. We can rephrase this as "Commercial and the logistic flows are coupled in the current system".

Standard journey of the products in the auction process is described in the following paragraph.

At least one night before the clock, growers fill an Electronic Delivery Form (EAB) which contains information about type, quality, length, volume of the products, whether they will be sold through FloraHolland Connect or the auction clock and to which

auction center the products will be delivered. Then, growers transport these specified products to the prespecified auction center before the clock either using their own truck or collective transporter. When arrived at the auction center, products are checked to see whether all supply information on EAB is correct. Moreover, trolleys are coupled with its content by a scan so as to follow the products by RFID throughout the auction center. Products that will be sold at the auction clock are transported to the quality control. After the control, they are sorted based on the product group and the clock they will be sold. They are stored at cold stores or delivery halls based on their specifications. Then, at 6:00 o'clock auction starts. During the auction, product trolleys are driven into the auction room on chain tracks. If the products are in large lots, only an example trolley is driven into the auction room and the rest is kept in the cold stores or delivery halls. Furthermore, sometimes only a picture of product is showed in the auction room instead of actual products, which is called image auctioning. The products are sold by a Dutch auction. Buyers can be physically present in the auction room or they can also buy the flowers via internet using a service called remote buying (Kopen Op Afstand, KOA). An Electronic Clock Transaction (EKT) is generated at the moment of the sale which contains product, quantity and buyer information. The quantity that a buyer can purchase is limited by the total supply quantity. Backordering is not allowed in an auction sale. After the auction process, sold products are transported to distribution rooms. About 95% of the time all supply is sold; however, 5% of the time some products are not sold. Unsold products are destroyed. Sold products are transferred from grower's containers to buyer's containers at distribution rooms. Finally, sold products are delivered to buyer's boxes located in the auction centers in FloraHolland network. Afterwards, sold products leave FloraHolland distribution network [43].

3.3 **Problem Definition**

Flows in FloraHolland network are introduced in Section 3.2.1. Dat [45] analyzed Clock Flows and Connect Flows for improvement possibilities and he observed some inefficiencies in Clock Flows. According to Dat [45], 70% of the products positioned to an auction center come from the hinterland of that auction center while 30% comes

from the hinterland of some other auction center. About 7% out of that 30% are transported to the auction center which is closest to the area where the products grew, after the auction. In other words, after the auction, about 7% of that 30% of products are transported back to the area where they came from before the auction. Moreover, some products are transported between more than one auction center which results in more transportation and handling costs and loss of time. In addition, the loss of time negatively affects the quality of the products because they are perishable.

These observations are the foundation of our case problem. To better understand the case problem, we can look at the example in Figure (3.2) by Gaki in [44]. A grower located close to Naaldwijk, decides to bring his/her products to Aalsmeer for the auction. After the auction, most of the products remains at the buyer boxes in Aalsmeer; however, some products need to be transferred back to Naaldwijk because their buyers' boxes are located there. These Naaldwijk \rightarrow Aalsmeer \rightarrow Naaldwijk flows are the inefficient flows observed in Clock flows. FloraHolland wants to avoid these unnecessary flows.

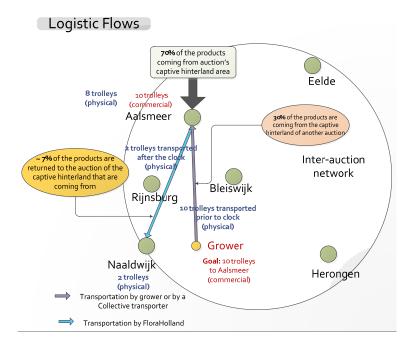


Figure 3.2: Current Case: Grower's Inventory Positioning [44]

Actually, these unnecessary flows are resulting from the coupling of logistic and commercial flows. Since the grower wants to attend the auction clock in Aalsmeer, all of his/her products need to be present in Aalsmeer in the current situation. Nevertheless,

boxes of the buyers wishing that product can be located at any of 6 auction centers. Thus, these products will be repositioned after the clock. Some will return back to Naaldwijk as in the example. However, if all of the products were not in Aalsmeer and we placed some of the products in Naaldwijk before the clock, we would avoid the inefficient Naaldwijk \rightarrow Aalsmeer \rightarrow Naaldwijk flow and we would save time and money. Consequently, it is crucial to decouple the commercial and the physical flows in order to avoid inefficient flows. What we suggest is that growers still can choose where their products will go through the clock process. However, the products are not obligated to be present at that auction center. Remember that products can be sold by image auctioning if there is not any product present in that auction center. Nonetheless, they still need to be present in the auction center network because type, quantity, length, volume and quality of the products should be checked before the auction. In other words, products need to be present at any auction center but not necessarily at the auction center that they will be auctioned prior to the clock. Moreover, products need to be physically positioned optimally within the auction center network prior to the clock so that growers and buyers can achieve less transportation costs, better service, and better quality for their products. The challenge here is that we do not know how many products will be purchased by whom and where the boxes of the buyers wishing that product are located until the auction.

Hence, now the question is *What is the optimal way in terms of cost to position the products initially in the auction center network prior to the clock, given the uncer-tainties of buyers and their purchase quantities?*

The uncertainties involved in the problem environment suggest using stochastic programming to formulate the model. Moreover, we know that these uncertainties are eliminated at the auction. After the auction, we know who bought how many products and where their boxes are located. This makes two-stage stochastic programming formulation an appropriate model for our case problem to answer the question above.

In our problem, first, inventory is allocated between the auction centers before the demand realization. Products are transported from the supplier to the auction centers at this stage. Then, after the demand realization, products are repositioned. At this stage, product movements are between the auction centers; which are the nodes of the

same echelon. Inventory movements between the locations of the same echelon are named as lateral transshipment in the literature. The case problem can also be studied as an inventory problem with lateral transshipments. As stated before, transportation between the auction centers occurs when the demand is revealed. In that sense, our problem resembles to reactive lateral transshipment. In the literature, reactive lateral transshipment is studied with periodic review and continuous review settings. However, in our problem, there is not any review. Unsold products are destroyed after the auctions. No inventory is hold for the next auction. Our problem is a single period problem. Reader is referred to [47] for more information about inventory problems with lateral transshipments.

In this study, we formulate the case problem as a two-stage stochastic problem. In the next chapter, two-stage stochastic programming formulation of the case problem will be explained in detail.

CHAPTER 4

THE RISK-NEUTRAL MODEL

In this study, the case problem mentioned in Chapter 3 is modeled with two risk preferences; risk-neutral and risk-averse. In this chapter, the risk-neutral model is presented. Mathematical model, solution approaches and their application to the model are explained in the following sections.

4.1 Risk-neutral Model

The case problem is constructed as a two-stage stochastic integer programming with recourse. Our aim is cost minimization by optimizing the initial product positioning prior to the clock in the auction center network. Who the buyers will be and their purchase quantities are the involved uncertainties. The minimum cost flow problem formulation without any limitation on capacity of arcs is used in the first and the second stages. In the first-stage, products are transported from the grower's location to the auction centers. They are positioned in the auction center network prior to the auction without any information of buyers and their purchase quantities. Second-stage starts with the auction process. After the auction, we know how many bouquets are bought by whom. We also know where these buyers' boxes are located. Hence, after the auction, we know where the bouquets should be delivered to. Our model is only for one grower. Moreover, it is not at buyer's level, it is at auction center level. We consolidate the buyers based on the location of their boxes. In other words, the individual demands of the buyers are not important for us. Instead, we are interested

in the demand¹ of the auction centers. Thus, after the auction, we know the demand of each auction center. Then, with this information, repositioning of the products inside the auction network takes place in the second-stage. In order to provide the feasibility in minimum cost flow formulation, total demand quantity should be equal to total supply quantity. In our problem environment, back-ordering is not allowed because the sale process is an auction. Hence, total demand of the auction centers cannot exceed the supply quantity. Most of the time, all supply is sold; hence, total demand equals to the supply. Nevertheless, there are rarely unsold products which are destroyed after the auction. Consequently, it is possible that supply quantity exceeds the total demand in the network. Considering this case, we introduce a dummy demand node to our model to provide feasibility.

A scenario based approach is used while formulating the problem as two-stage stochastic programming model. Main inputs of the model are the total supply quantity, unit traveling costs between the grower and the auction centers, unit traveling costs between each auction center and demand scenarios, occurrence probabilities of demand scenarios. Main outputs of the model are the initial product positioning decisions, optimal ship routes and volumes transported between the locations in the first-stage, optimal ship routes and volumes transported between the locations under each scenario in the second-stage and the corresponding first-stage transportation and expected second-stage transportation costs.

Formulation of the risk-neutral two-stage stochastic integer program (P1) is as follows:

¹ Note that due to the fact that sale process is an auction, actual demand and purchase quantities may not refer to the same quantities. In an auction, demand is a function of price. However, in this study, we assume that price does not influence the demand and the purchase quantity. When we say demand, we indeed refer to the purchase quantity which is assumed not to be affected by the price.

Sets

I, *J* set of nodes in the network

 $I, J = \{0, ..., 7\}$ I, J = 0 for the grower node I, J = 1, ..., 6 for the auction centersI, J = 7 for the dummy demand node

S set of scenarios

$$S = \{1, ..., S\}$$

Parameters

- c_{ij} unit cost per bouquet transported from node $i \in I$ to node $j \in J$ $c_{i7} = 0, c_{7j} = M$; node 7 is the dummy demand node
- sup number of bouquets supplied to the auction by the grower
- d_{is} demand of node $i \in \{1, ..., 7\}$ in terms of number of bouquets under scenario $s \in S$
- p_s probability of scenario $s \in S$

Decision Variables

- x_{ij} number of bouquets transported from node $i \in \{0, ..., 6\}$ to node $j \in \{1, ..., 6\}$ during initial product positioning (1st stage)
- b_i Initial inventory (number of bouquets) at node $i \in \{1, ..., 6\}$ after 1^{st} stage prior to the auction
- y_{ijs} number of bouquets transported from node $i \in \{1, ..., 7\}$ to node $j \in \{1, ..., 7\}$ during repositioning (2nd stage) under scenario $s \in S$
- **P1**

Minimize
$$\sum_{i=0}^{6} \sum_{j=1}^{6} c_{ij} x_{ij} + \sum_{s=1}^{S} p_s \sum_{i=1}^{7} \sum_{j=1}^{7} c_{ij} y_{ijs}$$
 (4.1)

subject to

$$\sum_{j=1}^{6} x_{0j} = sup$$

$$\sum_{j=1}^{6} x_{ji} - \sum_{i=1}^{6} x_{ij} = b_i$$

$$\forall i \in \{1, ..., 6\} \quad (4.3)$$

$$\sum_{j=1}^{7} y_{ijs} - \sum_{j=1}^{7} y_{jis} = b_i - d_{is} \qquad \forall i \in \{1, ..., 6\}, \forall s \in S \qquad (4.4)$$

$$\sum_{j=1}^{0} y_{7js} - \sum_{j=1}^{0} y_{j7s} = -d_{7s} \qquad \forall s \in S \qquad (4.5)$$

$$b_i \ge 0$$
, integer $\forall i \in I$ (4.6)

$$x_{ij} \ge 0, \text{integer}$$
 $\forall i \in \{0, ..., 6\}, \forall j \in \{0, ..., 6\}$ (4.7)

$$y_{ijs} \ge 0$$
, integer $\forall i \in \{1, ..., 7\}, \forall j \in \{1, ..., 7\}, \forall s \in S$ (4.8)

Objective function (4.1) minimizes the expected cost of transportation. The first term in the summation is the cost of the initial product positioning and the second term is the expected repositioning cost over all scenarios. Constraint (4.2) ensures that total number of bouquets that are transported from the grower to all auction centers is equal to the supply of the grower. In constraint set (4.3), the first term in left hand-side of the the equation is the total number of bouquets transported to that auction center (i.e. inflows). The second term in the left hand-side of the equation is the total number of bouquets transported from that auction center to the other auction centers (i.e. outflows). Note that when a product enters to FloraHolland network, it does not return back to the grower. The difference between the inflows and outflows at an auction center equals to the inventory at the auction center at the initial inventory positioning before the clock. Triangular inequality between the cost parameters cannot always hold. It might be cheaper to go from location A to location C through location B. Thus, we introduce constraint set (4.3) to the model. On the other hand, by manipulating the cost parameters (i.e. by setting $c_{AC} = c_{AB} + c_{BC}$ for the example above), it might be possible that constraint set (4.3) is not required anymore. However, for the general case, we keep constraint set (4.3) in the formulation. In constraint set (4.4) the first term in the equation is the total number of bouquets transported from that auction center to the other auction centers and to dummy node under scenario s (i.e. outflows). The second term in the equation is the total number of bouquets transported to that auction center under scenario s (i.e. inflows). Difference between the outflows and the inflows can be positive, negative or zero depending on the relation between the demand of the auction center under scenario s (d_{is}) and the initial inventory at the auction center (b_i).

- If the realized demand at the auction center under scenario *s* is higher than the initial inventory ($d_{is} > b_i$) then the difference between the demand and the initial inventory should be transported to that auction center during repositioning to satisfy the demand. Thus, the difference between the demand and the initial inventory should be equal to the inflows at the second-stage minus outflows at the second-stage.
- If the realized demand at the auction center under scenario *s* is lower than the initial inventory ($d_{is} < b_i$), it means that demand is not satisfied at some other auction center(s) or at the dummy demand node. Then the difference between the initial inventory and the demand should be transported from that auction center to the nodes that are lack of supply. Thus, the difference between the initial inventory and the demand should be equal to the outflows at the second-stage minus inflows at the second-stage.
- If the realized demand at the auction center under scenario *s* is equal to the initial inventory ($d_{is} = b_i$), the auction center can only act as a transshipment point. Thus, the outflows at the second-stage should be equal to the inflows at the second-stage.

Constraint (4.5) is fundamentally constraint (4.4) specialized for the dummy demand node. Because there is no initial inventory at the dummy demand node, the inflows minus outflows is set equal to the demand at the dummy demand node. Constraint sets (4.6), (4.7) and (4.8) are non-negativity and integrality constraints. FloraHolland works with both small and big growers. For a small grower, which sells small number of products at the auctions, it is important to find the exact integer number of products to position at the auction centers. This is why we keep integrality constraints in our formulation.

Feasibility Requirement

$$\sum_{i=1}^{7} d_{is} = sup \quad \forall s \in S$$

In order to provide feasibility, total demand of the nodes (i.e. auction centers and the dummy demand node) under each scenario should be equal to the total supply quantity.

There are 64s + 72 integer variables and 7s + 7 constraints in the formulation.

4.2 Solution Approaches for the Risk-neutral Model

In the previous section, we model the case problem as two-stage stochastic integer problem. Stochastic programs and integer programs are difficult to solve. Our model is both stochastic and integer. The proposed P1 in Section 4.1 can be solved by Cplex. However, for larger problems (i.e. more auction centers, higher number of demand scenarios) it may not be efficient to solve it by Cplex. For larger problems, it is crucial to use algorithms that benefit from the special structure of the model. P1 is a two-stage stochastic integer program with finite scenarios and the formulation is in extensive form. L-shaped method is the most frequently used method for large problems in extensive form [1]. As it can be seen from the literature review, almost all solution methods of two-stage stochastic integer programming are based on L-shaped method. Consequently, we propose L-shaped and Multicut L-shaped methods for our model, we present the general algorithms for L-shaped method and Multicut L-shaped method to solve the extensive form of the two-stage stochastic programming model (2.6) in Chapter 2.

4.2.1 L-shaped Method

L-shaped method by Van Slyke and Wets [4] can be seen as an extension of Bender's decomposition [5]. Different than Bender's decomposition, it also considers the feasibility questions in the stochastic programming. L-shaped method is a cutting plane

technique based on building an outer linearization of the recourse function $(\mathfrak{Q}(x))$ and a solution of the first-stage problem plus this linearization. L-shaped method is used to approximate the nonlinear recourse function. Recourse function involves a solution of all second-stage linear recourse programs. In the L-shaped method, instead of evaluating numerous functions, the recourse function is used to form a master problem in *x* and it is exactly evaluated only as a subproblem. A solution of the first-stage problem plus the linearization of recourse function form the master problem. Master problem finds a proposal *x* and sends it to the second-stage subproblems. Then, feasibility cuts which determine $\{x|Q(x) < +\infty\}$ and optimality cuts, which are the linear approximations of \mathfrak{Q} in its domain of finiteness, are sequentially added [1].

L-shaped method algorithm [1] for problem (2.6) in Chapter 2 is as follows: Step 0. Set r = k = v = 0.

Step 1. Set v = v + 1. Solve the master problem

Minimize
$$z = c^T x + Q$$

subject to $Ax = b$
 $D_l x \ge d_l$ $l = 1, ..., r$ (4.9)
 $E_l x + Q \ge e_l$ $l = 1, ..., k$ (4.10)
 $x \ge 0$
 Q , free variable

where constraint set (4.9) represents feasibility cuts and constraint set (4.10) represents optimality cuts. We set Q^{ν} equal to $-\infty$ initially and delete it from the computation if there is not any optimality cut in the master problem. Let (x^{ν}, Q^{ν}) be an optimal solution of the master problem.

Step 2. Check whether x^{ν} is second-stage feasible. For s = 1, ..., S, solve the following linear problem until w' > 0 for a scenario s.

Minimize
$$w' = e^T v^+ + e^T v^-$$

subject to $Wy + Iv^+ - Iv^- = h_s - T_s x^v$
 $y \ge 0, \quad v^+ \ge 0, \quad v^- \ge 0$

where $e^T = (1, ..., 1)$. If w' > 0 for a scenario, define

$$D_{r+1} = (\sigma_s^v)^T T_s,$$
$$d_{r+1} = (\sigma_s^v)^T h_s,$$

where σ_s^v is the associated simplex multiplier vector. Set r = r + 1, add the feasibility cut and return to Step 1. If for all scenarios, w' = 0, go to Step 3. *Step 3.* For each scenario $s \in \{1, ..., S\}$, solve the linear problem

Minimize
$$w = q_s^T y$$

subject to $Wy = h_s - T_s x^v$ (4.11)
 $y \ge 0$

Define,

$$E_{k+1} = \sum_{s=1}^{S} p_s(\pi_s^v)^T T_s,$$

$$e_{k+1} = \sum_{s=1}^{S} p_s(\pi_s^v)^T h_s,$$

$$w^v = e_{k+1} - E_{k+1} x^v$$

where π_s^v is the simplex multiplier vector of the optimal solution to the linear problem above for scenario s.

Stop if $Q^{\nu} \ge w^{\nu}$. x^{ν} is an optimal solution. Otherwise, set k = k+1, add the optimality cut, and return to Step 1.

4.2.2 Multicut L-shaped Method

In the L-shaped method, simplex multipliers of optimal solutions of the second-stage problems for all scenarios are aggregated to obtain one optimality cut in Step 3. Different than the L-shaped method, in Multicut L-shaped method these simplex multipliers are used to obtain one optimality cut per scenario, as necessary, in Step 3. Multicut L-shaped algorithm [1], [48] for problem (2.6) in Chapter 2 is as follows: **Step 0.** Set $r = v = 0, k_s = 0 \quad \forall s \in \{1, ..., S\}$. *Step 1.* Set v = v + 1. Solve the master problem

Minimize
$$z = c^T x + \sum_{s=1}^{S} Q_s$$

subject to $Ax = b$
 $D_l x \ge d_l$
 $l = 1, ..., r$
 $E_{l(s)}x + Q_s \ge e_{l(s)}$
 $x \ge 0$
 Q_s , free variable $\forall s$

Set Q_s^{ν} equal to $-\infty$ initially and delete it from the computation if there is not any optimality cut in the master problem. Let $(x^{\nu}, Q_1^{\nu}, ..., Q_s^{\nu})$ be an optimal solution of the master problem.

Step 2. Same with L-shaped method.

Step 3. For each scenario $s \in \{1, ..., S\}$, solve the linear problem (4.11). If

$$Q_s^v < p_s(\pi_s^v)^T (h_s - T_s x^v)$$
(4.12)

Define,

$$E_{k_s+1} = p_s(\pi_s^{\nu})^T T_s,$$
$$e_{k_s+1} = p_s(\pi_s^{\nu})^T h_s,$$

where π_s^v is the simplex multiplier vector of the optimal solution to (4.11) under scenario *s*. Set $k_s = k_s + 1$. If equation (4.12) does not hold for any scenario *s*, stop. x^v is an optimal solution. Otherwise, return to Step 1.

When L-shaped and multicut L-shaped methods are compared, it can be said that there is a trade off between them. By adding multicuts, more information is returned to the master problem. Thus, it is expected to find the optimal solution in less iterations. However, adding multiple cuts also mean larger master problems. The choice between one cut and multiple cuts may depend on the problem. It is expected that multicut method is more effective when the number of realizations is not significantly larger than the number of first-stage constraints.

4.3 Application of Solution Approaches to P1

In this section, application of L-shaped and multicut L-shaped methods to P1 is presented. In sections 4.2.1 and 4.2.2 classical L-shaped and multicut L-shaped method for two-stage stochastic linear programming is introduced. Both methods utilize the linearity of second-stage subproblem to generate optimality cuts by using the simplex multipliers of the optimal solution of the linear second-stage subproblem. In other words, optimality cuts in the classical L-shaped and multicut L-shaped method are based on duality theory in linear programming. Although we formulate our problem as two-stage stochastic integer program, we can still use the optimality cut generation methods in the classical L-shaped and multicut L-shaped method because secondstage subproblem of P1 is totally unimodular. Moreover, when the second-stage subproblem is totally unimodular, the optimal solution of the LP relaxation is integer and is the same with the optimal solution of the integer problem. Consequently, no additional method is required to obtain second-stage integer optimal solutions and to improve the optimality cuts. We benefit from the total unimodularity of the subproblem.

In both L-shaped and multicut L-shaped methods, the second-stage subproblem of P1 is the same. Its formulation is as follows;

Second-stage Subproblem

Minimize
$$\sum_{i=1}^{7} \sum_{j=1}^{7} c_{ij} y_{ij}$$

subject to
$$\sum_{j=1}^{7} y_{ij} - \sum_{j=1}^{6} y_{ji} = b_i - d_i \qquad \forall i \in \{1, ..., 6\}$$

$$\sum_{j=1}^{6} y_{7j} - \sum_{j=1}^{6} y_{j7} = -d_7$$

$$y_{ij} \ge 0, \text{ integer} \qquad \forall i \in \{1, ..., 7\}, \forall j \in \{1, ..., 7\}$$

As it can be seen from the formulation above, second-stage subproblem has minimum cost flow formulation; which is known to be totally unimodular. There is an extra b_i , $i \in 1, ..., 6$ parameter in the right hand side of the suproblem compared to the

classical formulation of minimum cost flow formulation. Thus, we need to check whether it violates the integrality of right hand side vector. In the L-shaped and multicut L-shaped methods, the master problem finds a proposal *x* and sends it to the second-stage subproblems. b_i , $i \in 1, ..., 6$, which are first-stage decision variables, are that proposals sent to our second-stage subproblem. Because b_i , $i \in 1, ..., 6$ have integrality restrictions in the first-stage problem, they are just integer parameters for the subproblem. Consequently, the right hand side vector $\begin{pmatrix} b-d \\ -d_7 \end{pmatrix}$ with $b = (b_1, ..., b_6)^T$, $d = (d_1, ..., d_6)^T$ is still integer. Hence, we can conclude that second-stage subproblem of P1 is totally unimodular.

4.3.1 Application of L-shaped Method to P1

In this section, the L-shaped algorithm adapted to P1 is illustrated. Notations are based on [1].

Step 0. Set k = v = 0. Step 1. Set v = v + 1. Solve the master problem

Master Problem

Let $(b^{\nu}, Q^{\nu}, x^{\nu})$ be an optimal solution. If no optimality cut constraint is present, Q^{ν} is set equal to $-\infty$ and is not considered in the computation of b^{ν} and x^{ν} .

Step 2. This step can be skipped because subproblem is always feasible for the current

optimal b^{ν} under all scenarios. Subproblem is a minimum cost flow problem and it is guaranteed that

$$\sum_{i=1}^{7} d_{is} = \sum_{i=1}^{6} b_i = sup \quad \forall s \in \{1, ..., S\}$$

Step 3. For each scenario $s \in \{1, ..., S\}$, solve the dual of the LP relaxation of subproblem.

LP relaxation of Primal Subproblem

Minimize
$$\sum_{i=1}^{7} \sum_{j=1}^{7} c_{ij} y_{ij}$$

subject to
$$\sum_{j=1}^{7} y_{ij} - \sum_{j=1}^{6} y_{ji} = b_i - d_i \qquad \forall i \in \{1, ..., 6\}$$
$$\sum_{j=1}^{6} y_{7j} - \sum_{j=1}^{6} y_{j7} = -d_7$$
$$y_{ij} \ge 0 \qquad \forall i \in \{1, ..., 7\}, \forall j \in \{1, ..., 7\}$$

Dual of LP relaxation of Subproblem

Maximize
$$\sum_{i=1}^{6} \pi_{i}(b_{i} - d_{i}) + \pi_{7}(-d_{7})$$

subject to $\pi_{i} - \pi_{j} \leq c_{ij}$ $\forall i \in \{1, ..., 7\}, \forall j \in \{1, ..., 7\}$
 π_{i} , free variable $\forall i \in \{1, ..., 7\}$ (4.14)

Technology matrix T of the LP relaxation of the subproblem is fixed for each scenario and T is equal to

$$\begin{bmatrix} -\mathbb{I}_6\\ 0 \end{bmatrix}$$

where $\mathbb I$ is the identity matrix.

Furthermore, h_s of the LP relaxation of subproblem equals

$$\begin{bmatrix} -d_{1s} \\ -d_{2s} \\ -d_{3s} \\ -d_{4s} \\ -d_{5s} \\ -d_{6s} \\ -d_{7s} \end{bmatrix}$$

Then, define

$$E_{k+1,i} = -\sum_{s=1}^{S} p_s \pi_{is}^{v} \quad \forall i \in \{1, ..., 6\}$$
$$e_{k+1} = \sum_{s=1}^{S} p_s \sum_{i=1}^{7} \pi_{is}^{v} (-d_{is})$$
$$w^{v} = e_{k+1} - \sum_{i=1}^{6} E_{k+1,i} b_{i}^{v}$$

If $Q^{\nu} \ge w^{\nu}$, stop; (b^{ν}, x^{ν}) is an optimal solution. Otherwise, set k = k + 1, add the optimality cut to the master problem and return to Step 1.

4.3.2 Application of Multicut L-shaped Method to P1

In this section, the multicut L-shaped algorithm adapted to P1 is illustrated. Notations are based on [1].

Step 0. Set v = 0 and $k_s = 0 \quad \forall s \in \{1, ..., S\}$. Step 1. Set v = v + 1. Solve the master problem

Master Problem

Let $(b^{\nu}, x^{\nu}, Q_1^{\nu}, ..., Q_s^{\nu})$ be an optimal solution. If there is not an optimality cut constraint in the master problem, Q_s^{ν} is set equal to $-\infty$ for all *s* and is not considered in the computation of b^{ν} and x^{ν} .

Step 2. Again this step can be skipped because subproblem is always feasible for the current optimal b^{ν} for all *s*.

Step 3. For each scenario $s \in \{1, ..., S\}$, solve (4.14). Let π_{is}^{v} be the associated simplex multipliers of optimal solution of (4.14). If

$$Q_s^v < p_s \sum_{i=1}^6 \pi_{is}^v (b_i - d_{is}) + \pi_7 (-d_{7s})$$
(4.15)

Define

$$E_{k_s+1,i} = -p_s \pi_{is}^{\nu} \quad \forall i \in \{1, ..., 6\}$$
$$e_{k_s+1} = p_s \sum_{i=1}^{7} \pi_{is}^{\nu} (-d_{is})$$

Set $k_s = k_s + 1$. If equation (4.15) does not hold for any scenario *s*, stop. (b^v , x^v) is an optimal solution. Otherwise, add the optimality cuts and return to Step 1.

CHAPTER 5

THE RISK-AVERSE MODEL

In the previous chapter, the case problem is modeled with risk-neutral preference. Now we model the same problem with risk-averse approach. Due to its computational advantages in optimization problems, we prefer first-order mean-semideviation as the risk measure. Moreover, because the case problem is a minimization problem, we use upper semideviation to measure the risk. In Section 5.1, the risk-averse model is presented. Objective function of the model is nonlinear and nondifferentiable. In Section 5.2, linearization of the risk-averse formulation and a decomposition based solution algorithm to solve the linearized formulation are described. Afterwards, in Section 5.3, linearization and decomposition method are applied to the case problem given.

5.1 The Risk-averse Model

As it is explained in literature review, mean-semideviation is one of the common mean-risk measures. The representation of the mean-semideviation measure is as follows [35]:

$$g_{\kappa,\delta_p} = \mathbb{E}[Y] + \kappa \delta_p[Y],$$

where $\delta_p[Y] = (\mathbb{E}[(Y - \mathbb{E}[Y])_+^p])^{\frac{1}{p}}$ is p^{th} central semideviation and κ is a nonnegative weight to trade off expected cost with risk.

In the context of stochastic programming problem of the form

min
$$\{\mathbb{E}[f(x,\omega)]: x \in X\},\$$

where $f(x, \omega) = c^T x + Q(x, \xi(\omega))$ and $Q(x, \xi(\omega))$ is the second-stage value function for a given realization ω , the mean-semideviation objective function takes the form [35]:

$$g_{\kappa,\delta_p}(f(x,\omega)) = \mathbb{E}[f(x,\omega)] + \kappa (\mathbb{E}[(f(x,\omega) - \mathbb{E}[f(x,\omega)])_+^p])^{\frac{1}{p}}.$$

Moreover, for p = 1, semideviation function can be written as

$$\delta(f(x,\omega)) = \mathbb{E}\left[(f(x,\omega) - \mathbb{E}[f(x,\omega)])_+\right] = \mathbb{E}\left[\max\{f(x,\omega) - \mathbb{E}[f(x,\omega)], 0\}\right]$$

and mean-semideviation objective function becomes

$$g_{\kappa,\delta} = \mathbb{E}[f(x,\omega)] + \kappa\delta(f(x,\omega)) = \mathbb{E}[f(x,\omega)] + \kappa\mathbb{E}\left[\max\{f(x,\omega) - \mathbb{E}[f(x,\omega)], 0\}\right]$$

By straight forward calculations in [41],

$$g_{\kappa,\delta} = \mathbb{E}[f(x,\omega)] + \kappa \mathbb{E}\left[\max\{f(x,\omega) - \mathbb{E}[f(x,\omega)], 0\}\right]$$
$$= (1-\kappa)\mathbb{E}[f(x,\omega)] + \kappa \mathbb{E}\left[\max\{f(x,\omega), \mathbb{E}[f(x,\omega)]\}\right].$$
(5.1)

With the assumptions that ξ is discrete and has finite support $\{\xi_1, ..., \xi_S\}$ with corresponding probabilities $p_1, ..., p_S$, using (5.1), two-stage stochastic problem with mean-semideviation risk measure takes the form [37]:

$$\min\left\{c^{T}x + (1-\kappa)\sum_{s=1}^{S} p_{s}Q(x,\xi_{s}) + \kappa\sum_{s=1}^{S} p_{s}\max\{Q(x,\xi_{s}),\sum_{\bar{s}=1}^{S} p_{\bar{s}}Q(x,\xi_{\bar{s}})\} : x \in X\right\}$$
(5.2)

where $Q(x, \xi_s) = \min_{y_s} \{q_s^T y_s | W_s y_s = h_s - T_s x, y_s \ge 0\}$ is the second-stage value function.

Formulation (5.2) given above is applied to our case problem in order to include a risk factor into the objective. We formulate the risk-averse model as a mean-semideviation model with p = 1, $\kappa = 1$ and finite discrete ξ as follows:

Minimize
$$\sum_{i=0}^{6} \sum_{j=1}^{6} c_{ij} x_{ij} + \sum_{s=1}^{S} p_s \max \left\{ Q(b,\xi_s), \sum_{\bar{s}=1}^{S} p_{\bar{s}} Q(b,\xi_{\bar{s}}) \right\}$$

subject to

$$\sum_{j=1}^{6} x_{0j} = sup$$

$$\sum_{j=0}^{6} x_{ji} - \sum_{j=1}^{6} x_{ij} = b_i \quad \forall i \in \{1, ..., 6\}$$

$$b_i \ge 0, \text{integer} \quad \forall i \in \{1, ..., 6\}$$

$$x_{ij} \ge 0, \text{integer} \quad \forall i \in \{0, ..., 6\}, \forall j \in \{0, ..., 6\}$$

where

$$Q(b,\xi_s) = Min \qquad \sum_{i=1}^{7} \sum_{j=1}^{7} c_{ij} y_{ijs}$$

subject to

$$\sum_{j=1}^{7} y_{ijs} - \sum_{j=1}^{6} y_{jis} = b_i - d_{is} \qquad \forall i \in \{1, ..., 6\}, \forall s \in S$$
$$\sum_{j=1}^{6} y_{7js} - \sum_{j=1}^{6} y_{j7s} = -d_{7s} \qquad \forall s \in S$$
$$y_{ijs} \ge 0, \text{integer} \qquad \forall i \in \{1, ..., 7\}, \forall j \in \{1, ..., 7\}, \forall s \in S$$

The extensive form of the formulation above, which is referred as MSD in the following paragraphs, is as follows :

Minimize

$$\sum_{i=0}^{6} \sum_{j=1}^{6} c_{ij} x_{ij} + \sum_{s=1}^{S} p_s \max\left\{\sum_{i=1}^{7} \sum_{j=1}^{7} c_{ij} y_{ijs}, \sum_{\bar{s}=1}^{S} p_{\bar{s}} \sum_{i=1}^{7} \sum_{j=1}^{7} c_{ij} y_{ij\bar{s}}\right\}$$
(5.3)

subject to

Constraint Sets (4.2), (4.3), (4.4), (4.5), (4.6), (4.7) and (4.8).

Feasibility requirement is the same with the risk-neutral model.

$$\sum_{i=1}^{7} d_{is} = sup \quad \forall s \in S$$

Constraint sets of risk-neutral and risk-averse model are the same. Thus, both models have the same feasible region. Furthermore, note that by constraint (4.6), (4.7) and (4.8), we keep the integrality restriction also for the risk-averse model. The only difference between risk-neutral and risk-averse models is in the objective function. In the risk-neutral model, the objective is to minimize the expected transportation cost in the network over all possible scenarios. However, in the risk-averse approach, we introduce a risk measure into the model. In the objective function, besides the expected transportation cost we also minimize expected excess from the expected cost. We do not know which demand scenario will occur after the clock; hence, we want to avoid the case that a scenario's cost has a really high cost compared to the expected cost.

5.2 Solution Approaches for Risk-averse Model

Objective function of MSD is both nonlinear and nondifferentiable. In this subsection, we present the linearization of this type of objective function in general notations. In other words, we explain how to linearize formulation (5.2). Then, we describe a variation of multicut L-shaped method to solve a linearized version of formulation (5.2). Application of the solution methods proposed for the case problem will be covered in the following section.

When we look at formulation (5.2), max $\{Q(x,\xi_s), \sum_{\bar{s}=1}^{\bar{s}} p_{\bar{s}}Q(x,\xi_{\bar{s}})\}$ part of the objective function causes the nonlinearity. It is handled by additional constraints (5.4), (5.5), (5.6), (5.7). t_s is forced to take the maximum value of $\{Q(x,\xi_s), \sum_{\bar{s}=1}^{\bar{s}} p_{\bar{s}}Q(x,\xi_{\bar{s}})\}$. Consequently, formulation (5.2) is equivalently written as [37]

Minimize
$$c^T x + (1 - \kappa) \sum_{s=1}^{S} p_s \theta_s + \kappa \sum_{s=1}^{S} p_s t_s$$

subject to

$$Ax = b,$$

$$W_s y_s = h_s - T_s x \qquad \forall s \in S,$$

$$q_s^T y_s \le \theta_s, \qquad \qquad \forall s \in S \qquad (5.4)$$

$$\sum_{\bar{s}}^{\bar{s}} p_{\bar{s}} \theta_{\bar{s}} \le t_s, \qquad \qquad \forall s \in S \qquad (5.5)$$

$$\theta_s \le t_s, \qquad \qquad \forall s \in S \qquad (5.6)$$

$$x \ge 0, \quad y_s \ge 0 \qquad \qquad \forall s \in S$$

$$t_s, \theta_s$$
, free variables, $\forall s \in S$ (5.7)

Linearized version of model (5.2) can be solved by a linear programming solver. Nonetheless, in the case problem, we work with integer variables. Thus, model (5.2)with integrality constraints may not be efficiently solvable for large problems. This exposes the need of decomposition algorithms for large problems. Model 5.2 does not have a block structure that fits to the existing stochastic programming schemes either. The problem comprises explicit coupling between scenario dependent variables. Consequently, most known algorithms do not work for the problem. Kristoffersen [37] introduces a decomposition algorithm for model 5.2. Kristoffersen [37] treats some scenario dependent variables as first-stage variables; which enables a certain degree of separability and so a scenario wise cutting plane algorithm. The decomposition algorithm in [37] is akin to multicut L-shaped algorithm. The idea behind the method is to relax $Q(x,\xi_s) < +\infty$, which are so-called induced constraints, and the $Q(x,\xi_s) \leq \theta_s$ constraints for $\forall s \in S$ and then iteratively strengthen them with feasibility and optimality cuts. In [37], the method is explained for mean-central deviation model; nonetheless the algorithm is stated to work for mean-semideviation as well. We illustrate decomposition method in [37] for mean-semideviation using the notation of L-shaped method given in [1].

Kristoffersen's Decomposition Method for mean-semideviation two-stage stochastic linear program is as follows:

Step0.Set $r = v = 0, k_s = 0 \quad \forall s \in \{1, ..., S\}$.

Step1. Set v = v + 1. Solve the current master problem

Minimize
$$c^T x + (1 - \kappa) \sum_{s=1}^{S} p_s \theta_s + \kappa \sum_{s=1}^{S} p_s t_s$$

subject to
 $\sum_{\overline{s}}^{S} p_{\overline{s}} \theta_{\overline{s}} \le t_s,$ $\forall s \in S$
 $\theta_s \le t_s,$ $\forall s \in S$
 $x \in X,$
 $D_l x \ge d_l$ $l = 1, ..., r$
 $E_{l(s)} x + \theta_s \ge e_{l(s)}$ $l(s) = 1, ..., k_s$
 t_s, θ_s , free variables, $\forall s \in S$

Let $x^{\nu}, t_1^{\nu}, ..., t_s^{\nu}, \theta_1^{\nu}, ..., \theta_s^{\nu}$ be an optimal solution. (If $t_s^{\nu} = -\infty$ and $\theta_s^{\nu} = -\infty$ for some $s \in \{1, ..., S\}$, ignore them in the computation.)

Step2. This step is checking second-stage feasibility of x^{ν} . Second-stage subproblems are the same as the L-shaped method subproblems (4.11). Thus, this step is the same with Step 2 of the L-shaped method. Check whether x^{ν} is second-stage feasible. For s = 1, ..., S, solve the linear problem until w' > 0 for a scenario s.

Minimize
$$w' = e^T v^+ + e^T v^-$$

subject to $Wy + Iv^+ - Iv^- = h_s - T_s x^V$
 $y \ge 0, \quad v^+ \ge 0, \quad v^- \ge 0$

where $e^T = (1, ..., 1)$. If w' > 0 for a scenario, define

$$D_{r+1} = (\sigma_s^v)^T T_s,$$
$$d_{r+1} = (\sigma_s^v)^T h_s,$$

where σ_s^v is the associated simplex multiplier vector. Set r = r + 1, add the feasibility cut and return to Step 1. If for all scenarios, w' = 0, go to Step 3.

Step3. For each scenario $s \in \{1, ..., S\}$, solve the linear problem (4.11) with $x = x^{\nu}$. If

$$(\pi_s^{\nu})^T (h_s - T_s x^{\nu}) > \theta_s^{\nu}, \tag{5.8}$$

 $..., k_s$

Define,

$$E_{k_s+1} = (\pi_s^v)^T T_s,$$
$$e_{k_s+1} = (\pi_s^v)^T h_s,$$

where π_s^{ν} is the simplex multiplier vector of the optimal solution of (4.11) for scenario *s*. Set $k_s = k_s + 1$. If equation (5.8) does not hold for any scenario *s*, stop. x^{ν} is an optimal solution. Otherwise, return to Step 1.

5.3 Application of the Solution Approaches to MSD

In the previous section, linearized equivalent model for model (5.2) is introduced. Now we present the linearized equivalent of MSD; which is referred as MSDL. After introducing MSDL, we explain how Kristoffersen's decomposition method is applied to MSDL. Note that, model (5.2) is a two-stage stochastic linear problem, which does not have any integrality restrictions. However, MSDL is a two-stage stochastic integer problem. Despite the integrality restriction, we still make use of the linearization method and the decomposition method suggested in the previous section.

Objective function (5.3) of MSD is nonlinear and nondifferentiable. We linearize the objective function and model turns into equivalent mixed integer program (MSDL):

Minimize
$$\sum_{i=0}^{6} \sum_{j=1}^{6} c_{ij} x_{ij} + \sum_{s=1}^{S} p_s t_s$$

subject to

$$\begin{split} \sum_{j=1}^{6} x_{0j} &= \sup p, \\ \sum_{j=0}^{6} x_{ji} - \sum_{j=1}^{6} x_{ij} &= b_i, \\ \sum_{j=1}^{7} y_{ijs} - \sum_{j=1}^{6} y_{jis} &= b_i - d_{is} \\ \sum_{j=1}^{7} y_{ijs} - \sum_{j=1}^{6} y_{j7s} &= -d_{7s} \\ \sum_{j=1}^{7} y_{7js} - \sum_{j=1}^{6} y_{7js} &= -d_{7s} \\ \sum_{i=1}^{7} \sum_{j=1}^{7} c_{ij}y_{ijs} &\leq \theta_s, \\ \sum_{i=1}^{7} \sum_{j=1}^{7} c_{ij}y_{ijs} &\leq \theta_s, \\ \sum_{i=1}^{8} p_i \theta_i &\leq t_s, \\ \theta_s &\leq t_s, \\ \theta_s &\leq t_s, \\ \theta_s &\leq t_s, \\ \psi_s &\in S \\ \theta_i &\geq 0, \text{integer}, \\ x_{ij} &\geq 0, \text{integer}, \\ \psi_i &\in \{1, ..., 7\}, \forall j \in \{1, ..., 7\}, \forall s \in S \\ y_{ijs} &\geq 0, \text{integer}, \\ \psi_s &\in S \\ \psi$$

Although MSDL can be solved by a commercial mixed integer programming (MIP) solver, we adopt Kristoffersen's algorithm to MSDL so as to solve larger problems. Kristoffersen [37] studies with two-stage stochastic linear programs. Like classical L-shaped method, Kristoffersen's decomposition method also benefits from duality theory to generate the optimality cuts using the simplex multipliers. In MSDL, some variables are restricted to be integers. Moreover, subproblem used in application of Kristoffersen's algorithm to MSDL is the same as the subproblem (4.13), all variables of which are integer. However, these integrality constraints do not bring about any problem. We again benefit from the total unimodularity of the subproblem (4.13). In addition, because the second-stage subproblem is totally unimodular, no additional

method is required to obtain integer optimal solutions for second-stage variables and no additional method is required to improve the optimality cuts because they are obtained by strong duality theory. We use LP relaxation of subproblem 4.13 to obtain the optimality cuts in Kristoffersen's decomposition algorithm.

The adaptation of Kristoffersen's decomposition method to MSDL is as follows:

Step 0. Set v = 0 and $k_s = 0 \quad \forall s \in \{1, ..., S\}$. Step 1. Set v = v + 1. Solve the master problem

ā

Minimize
$$\sum_{i=0}^{6} \sum_{j=1}^{6} c_{ij} x_{ij} + \sum_{s=1}^{5} p_s t_s$$
subject to
$$\sum_{i=0}^{6} x_{0i} = sup$$

$$\sum_{j=0}^{6} x_{ji} - \sum_{j=1}^{6} x_{ij} = b_i \qquad \forall i \in \{1, ..., 6\}$$

$$b_i \ge 0$$
, integer $\forall i \in \{1, ..., 6\}$

$$x_{ij} \ge 0$$
, integer $\forall i \in \{0, ..., 6\}, \forall j \in \{0, ..., 6\}$

$$\sum_{\bar{s}=1}^{S} p_{\bar{s}} \theta_{\bar{s}} \le t_s \qquad \qquad \forall s \in S$$

$$\theta_s \leq t_s$$

$$\sum_{i=1}^{6} E_{iter(s),i}b_i + Q_s \ge e_{iter(s)} \qquad iter(s) = 1, ..., k_s$$
$$t_s, \theta_s, \quad \text{free variables} \qquad \forall s \in S$$

 $\forall s \in S$

Solve the current master problem and let $b^v, x^v, t_1^v, ..., t_S^v, \theta_1^v, ..., \theta_S^v$ be an optimal solution. (For v = 1, set $t_s^v = -\infty$ and $\theta_s^v = -\infty \forall s \in \{1, ..., S\}$ and ignore them in the computation of b^v, x^v .)

Step2. This step can be skipped again as in the L-shaped algorithm since the sub-

problem is always feasible for the current optimal b^{ν} under all scenarios. We use the LP relaxation of the subproblem (4.13) as the subproblem. It is a minimum cost flow problem and it is guaranteed that

$$\sum_{i=1}^{7} d_{is} = \sum_{i=1}^{6} b_i = sup \quad \forall s \in \{1, ..., S\}$$

Step3. For each scenario $s \in \{1, ..., S\}$, solve the dual of the LP relaxation of the subproblem (4.13) with $b = b^{v}$.

Dual of LP relaxation of Subproblem

Maximize
$$\sum_{i=1}^{6} \pi_{i}(b_{i} - d_{i}) + \pi_{7}(-d_{7})$$

subject to $\pi_{i} - \pi_{j} \leq c_{ij}$ $\forall i \in \{1, ..., 7\}, \quad \forall j \in \{1, ..., 7\}$
 π_{i} , free variable $\forall i \in \{1, ..., 7\}$ (5.10)

 π is the dual variable vector.

Technology matrix T of the LP relaxation of subproblem is fixed for each scenario and T is equal to

$$\begin{bmatrix} -\mathbb{I}_6\\ 0 \end{bmatrix}$$

where \mathbb{I} is the identity matrix. Furthermore, h_s of the LP relaxation of subproblem equals

$$\begin{bmatrix} -d_{1s} \\ -d_{2s} \\ -d_{3s} \\ -d_{4s} \\ -d_{5s} \\ -d_{6s} \\ -d_{7s} \end{bmatrix}$$

$$\sum_{i=1}^{6} \pi_{is}^{\nu}(b_i - d_{is}) + \pi_{7s}^{\nu}(-d_{7s}) > \theta_s^{\nu}$$
(5.11)

for some *s*, then define

$$E_{k_s+1,i} = -\pi_{is}^{v} \quad \forall i \in \{1, ..., 6\}$$
$$e_{k_s+1} = \sum_{i=1}^{7} \pi_{is}^{v} (-d_{is})$$

Set $k_s = k_s + 1$. If equation (5.11) does not hold for any scenario *s*, stop. (b^{ν} , x^{ν}) is an optimal solution. Otherwise, add the optimality cuts and return to Step 1.

If

CHAPTER 6

COMPUTATIONAL STUDY

Computational study is divided into three main parts. In section 6.2, computational study of the risk-neutral model is given. The risk-neutral problem is formulated as two-stage stochastic integer programming model (P1). However, because this formulation has the complexities of both integer programming and stochastic programming, for larger problems L-shaped and multicut L-shaped decomposition methods are applied. In section 6.2 we compare computational efficiencies of the alternative solution methods for the risk-neutral problem and show the value of stochastic solution. In section 6.3, computational study of the risk-averse model is given. We compare computational efficiencies of MSDL and the proposed multicut decomposition based solution algorithm for it. In section 6.4, we compare the results of risk-neutral and risk-averse approaches. Before giving the computational results mentioned above, in section 6.1, we illustrate the computational design parameters which are common in both risk-neutral and risk-averse problems.

Mathematical models are coded in GAMS 23.7. Demand scenarios are generated in MATLAB R2013.a. GAMS is called from MATLAB and optimization is done by using CPLEX 12 solver. A PC with Intel(R) Core(TM) i5-4200M CPU 2.50GHz and 6 GB RAM running Windows 8 is used to run the codes.

6.1 Computational Design

In both risk-neutral and risk-averse approaches, the main inputs of the model are total supply quantity, unit traveling costs between locations, demand scenarios and occurrence probability of demand scenarios. Models are run with two different supply quantities; 100 and 1,000. Moreover, models are run with different number of scenarios. More scenarios mean that more information is given to the model. Unit traveling costs between locations are given by the company. In [44], Gaki also uses the same unit traveling costs, which are stated to be calculated considering route's distance, capacity utilization of trucks, average number of truck rides, fixed and variable transportation costs. Owing to the lack of demand information of auction centers, we generate demand scenarios. Each demand scenario is taken equally likely to occur.

We know that 95% percent of the time all the supply is sold and 5% of the time there are leftovers. Thus, we consider this reality while generating our demand scenarios. In a demand scenario set, we make sure that supply equals to demand in 95% of the scenarios and that supply exceeds demand in 5% of the scenarios.

For supply=demand case, following algorithm is used to generate demand for each auction center.

- Generate 5 random integers in [0, supply]
- Sort these 5 numbers in ascending order
- Add "0" at the beginning and add "supply" at the end of the array, hence we have 7 sorted numbers
- Take the differences of successive numbers
- Set demand of dummy demand node equal to 0

Because the last element in the array is the supply quantity, we make sure that total demand of the auction centers equal to supply.

For supply>demand case, following algorithm is used to generate demand for each auction center.

- Generate 6 random integers in [0, supply-1]
- Sort these 6 numbers in ascending order
- Add "0" at the beginning, hence we have 7 sorted numbers

- Take the differences of successive numbers
- Set demand of dummy demand node equal to "supply-total demand of auction centers"

Because the last element in the array can have a maximum value of "supply quantity-1", we make sure that total demand of auction centers is less than supply.

6.2 Results of the Risk-neutral Model

In this section, we solve P1 by CPLEX. Then, we also solve the LP relaxation of P1 by CPLEX. When we compare the results of P1 and the LP relaxation of P1, we observe that optimal solution values of them are the same. We computationally show that P1 is totally unimodular. Then, since it is easier to solve linear problems compared to the integer problems, we solve L-shaped decomposition and multicut L-shaped decomposition for the LP relaxation of P1 instead of P1. In other words, we relax the integrality restriction in the master problem of the decomposition methods. Different number of scenarios and two different supply quantities are used for the computational study.

Table 6.1 and Table 6.2 show the results of Cplex solution of P1 for different scenario numbers up to 65,000 scenarios for supply quantities 100 and 1,000 respectively. The results show that for both supply quantities, test problems generated for P1 can be solved by CPLEX solver for a number of scenarios up to 60,000. For both supply quantities, CPLEX elapsed time and number of CPLEX iterations to solve P1 increase with the increase in the number of scenarios as expected.

Number	Elapsed Time	Number of	Objective
Of Scenarios	(in sec.)	CPLEX Iterations	Function Value
500	0.764	3177	1944.836
1000	1.688	6430	1927.651
2500	7.848	16128	1954.115
5000	29.548	31538	1958.026
10000	129.866	66139	1948.589
25000	947.161	164850	1944.501
50000	4092.297	326789	1948.505
60000	6026.284	407545	1947.348
65000	No solution	No solution	No solution

Table6.1: CPLEX results of P1 with a supply quantity of 100

Table6.2: CPLEX results of P1 with a supply quantity of 1,000

Number	Elapsed time	Number of	Objective
Of Scenarios	(in sec.)	Cplex Iterations	Function Value
500	0.731	3229	19346.207
1000	1.693	6596	19033.914
2500	8.034	16587	19333.766
5000	30.592	32873	19448.594
10000	134.902	67268	19435.468
25000	1016.662	171593	19375.946
50000	4416.247	347098	19421.776
60000	6241.690	397373	19390.767
65000	No solution	No solution	No solution

As explained in Chapter 4, the minimum cost flow formulation is used in the first and the second stages of P1. It is known that minimum cost flow formulation is totally unimodular. However, having totally unimodular first and second stages does not guarantee that overall formulation is totally unimodular [46]. After solving P1 by CPLEX, we also solve the LP relaxation of P1 by CPLEX to see whether optimal values of the decision variables are the same. Table 6.3 and Table 6.4 demonstrate the elapsed times and objective function values for the LP relaxation of P1 for supply quantities 100 and 1,000 respectively.

Number Of Scenarios	Elapsed Time (in sec.)	Objective Function Value
500	0.672	1944.836
1000	1.478	1927.651
2500	5.424	1954.115
5000	16.845	1958.026
10000	71.375	1948.589
25000	494.370	1944.501
50000	1913.692	1948.505
75000	4273.849	1945.382
100000	8034.315	1943.293
125000	No solution	No solution

Table6.3: CPLEX results of the LP relaxation of P1 with a supply quantity of 100

Table6.4: CPLEX results of the LP relaxation of P1 with a supply quantity of 1,000

Number Of Scenarios	Elapsed Time (in sec.)	Objective Function Value
500	0.626	19346.207
1000	1.352	19033.914
2500	5.539	19333.766
5000	17.162	19448.594
10000	71.578	19435.468
25000	571.626	19375.946
50000	2380.826	19421.776
75000	4564.726	19412.097
100000	8128.364	19382.601
125000	No solution	No solution

Observation 6.2.1 Objective function values of the optimal solutions of the LP relaxation of P1 for all the test problems are the same as the objective function values of the optimal solutions of P1. P1 is computationally totally unimodular.

Based on the results above, for both supply quantities, the LP relaxation of P1 can be solved by CPLEX solver for a number of scenarios up to 100,000. CPLEX elapsed time to solve the LP relaxation of P1 raises when the number of scenarios increase.

As it is expected, the LP relaxation of P1 is computationally preferable to P1. While P1 can be solved by CPLEX up to 60,000 scenarios, the LP relaxation of P1 can be solved up to 100,000 scenarios. Moreover, elapsed times of LP relaxation of P1 are lower than elapsed times of P1.

After seeing that LP relaxation of P1 gives the same optimal results as P1, we change the setup of L-shaped and multicut L-shaped algorithms for the computational study. In Chapter 4, master problem of L-shaped method and multicut L-shaped method have the integrality restrictions. Now, we relax the integrality restrictions in the master problems and run the test scenarios by using LP relaxation of the master problems in the L-shaped and multicut L-shaped algorithms. Results are given in Table 6.5 and Table 6.6 for supply quantities 100 and 1,000.

Table6.5: L-shaped and multicut L-shaped results of the LP relaxation of P1 with a supply quantity of 100

	L-shaped	method	Mult	Multicut L-shaped method		
Number of Scenarios	Elapsed Time (in sec.)	Number of Iterations	Elapsed Time (in sec.)	Number of Iterations	Number of Cuts	Objective Function Value
500	3553.945	53	483.724	9	3663	1944.836
1000	6756.800	52	1351.358	11	8289	1927.651
2500	16973.344	52	3525.436	10	20545	1954.115
5000	42559.545	55	7633.628	11	41236	1958.026

Table6.6: L-shaped and multicut L-shaped results of the LP relaxation of P1 with a supply quantity of 1,000

	L-shaped	method	Mult	Multicut L-shaped method		
Number of Scenarios	Elapsed Time (in sec.)	Number of Iterations	Elapsed Time (in sec.)	Number of Iterations	Number of Cuts	Objective Function Value
500	4799.250	73	577.666	11	4142	19346.207
1000	9538.352	78	1387.954	11	8205	19033.914
2500	25260.411	81	3718.809	11	20722	19333.766
5000	59832.890	82	7950.325	12	41043	19448.594

Table 6.5 and Table 6.6 show that elapsed time of L-shaped algorithm increases with the increasing number of scenarios; which is reasonable.

Observation 6.2.2 Problems with supply quantity of 1,000 have higher elapsed times than the problems with supply quantity of 100 given that scenario sizes are equal. This is probably related with the observation that problems with supply quantity of 1,000 have higher number of iterations than problems with supply quantity of 100. This is because, the number of subproblems solved at each iteration is equal to the number of scenarios.

Observation 6.2.3 Elapsed times and number of cuts of the multicut L-shaped algorithm raise if the scenario sizes increase. On the other hand, number of iterations does not have any increasing or decreasing trend.

Observation 6.2.4 For both supply quantities, the multicut L-shaped algorithm is more efficient than the L-shaped algorithm in terms of elapsed time for solving the test problems.

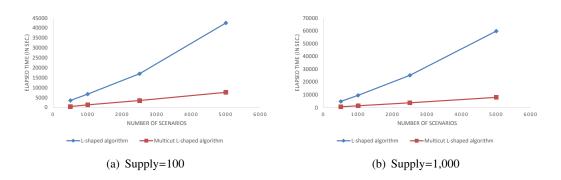


Figure 6.1: Comparison of elapsed times of L-shaped and multicut L-shaped algorithms

Observation 6.2.5 Elapsed time of L-shaped method with a scenario size 5,000 is higher than the elapsed time of the CPLEX solution of the LP relaxation of P1 with a scenario size 100,000. Furthermore, elapsed time of the multicut L-shaped algorithm with a scenario size 5,000 is almost equal to the elapsed time of the CPLEX solution of the LP relaxation of P1 with a scenario size 100,000. In addition, we observe that elapsed times increase with the increase in the number of scenarios. Consequently, it can be said that L-shaped and multicut L-shaped methods are not preferable for the problems that CPLEX can solve. They may be useful only when CPLEX cannot solve a problem.

Stochastic programs are known to be computationally hard to solve. In real life, people tend to solve simpler models to answer the same questions instead of using stochastic programs. For example, random variables are replaced with their expected values to solve the simpler deterministic problem. This problem is called the expected value problem, and its solution is named as the expected value solution. Difference between the expected result of the stochastic solution ($\mathbb{E}[SS]$) and the expected result of using the expected value solution as the first stage decision ($\mathbb{E}[EVS]$) is called the value of stochastic solution (VSS). Reader is referred to [1] for more information. Table 6.7 and Table 6.8 demonstrate the value of stochastic solution for supply quantities 100 an 1,000 respectively.

Number of Scenarios	E[EVS]	E[SS]	VSS	Decrease in Percentages
500	2276.598	1944.836	331.762	14.57%
1000	2254.607	1927.651	326.956	14.50%
2500	2304.788	1954.115	350.673	15.21%
5000	2311.948	1958.026	353.922	15.31%
10000	2306.394	1948.589	357.805	15.51%
25000	2307.736	1944.501	363.235	15.74%
50000	2309.237	1948.505	360.732	15.62%
75000	2308.873	1945.382	363.491	15.74%
100000	2309.513	1943.293	366.220	15.86%

Table6.7: Value of stochastic solution for a supply quantity 100

Table6.8: Value of stochastic solution for a supply quantity 1000

Number of Scenarios	𝔅 [EVS]	E[SS]	VSS	Decrease in Percentages
500	22895.998	19346.207	3549.791	15.50%
1000	22464.937	19033.914	3431.023	15.27%
2500	22720.562	19333.766	3386.796	14.91%
5000	22947.060	19448.594	3498.466	15.25%
10000	22878.982	19435.468	3443.514	15.05%
25000	22845.825	19375.946	3469.879	15.19%
50000	22854.376	19421.776	3432.600	15.02%
75000	22883.691	19412.097	3471.594	15.17%
100000	22822.714	19382.601	3440.113	15.07%

Observation 6.2.6 Solving the stochastic problem instead of using the expected value solution as the first stage decision results in about 15% decrease in the expected total cost.

6.3 Results of the Risk-averse Model

In this section, we compare the computational efficiencies of MSDL and the proposed multicut decomposition algorithm. Several problems with different number of scenarios and 2 different supply quantities are used to run the alternative methods. Table 6.9 and Table 6.10 show the elapsed time to solve each problem, number of Cplex iterations, number of multicut algorithm iterations, number of cuts and objective function results for supply quantities 100 and 1,000 respectively.

	C	PLEX	Mul	Multicut Algorithm	n	
Number of	Elapsed	Number	Elapsed	Number	Number	Objective
Scenarios	Time (in sec.)	of Cplex Iteration	Time (in sec.)	of Iterations	of Cuts	Function Value
100	0.528	2883	124.343	11	851	2084.765
250	4.519	8070	274.676	10	1928	2136.761
500	13.801	17372	815.404	12	3950	2106.729
750	34.590	28851	1275.190	10	5773	2122.126
1000	184.344	41652	1549.639	10	7963	2133.296
2000	1108.126	90738	2986.416	10	16547	2099.785
4000	12661.447	193775	9023.048	10	31804	2121.424
4500	No solution	No solution	14419.43	12	37409	2112.774
5000	No solution	No solution	No solution	No solution	No solution	No solution

Table 6.9: Comparison of CPLEX solver and multicut algorithm for risk-averse model with a supply quantity of 100

	C	CPLEX	Mul	Multicut Algorithm	n	
Number of	Elapsed	Number	Elapsed	Number	Number	Objective
Scenarios	Time (in sec.) of	of Cplex Iteration	Time (in sec.)	of Iterations	of Cuts	Function Value
100	0.586	4295	142.620	11	875	21246.231
250	1.999	9689	393.979	12	2209	20946.373
500	11.847	20116	739.321	11	4138	21349.315
750	58.614	31473	1135.908	11	6329	20805.031
1000	134.456	46323	1545.488	11	8428	20792.810
2000	1345.481	97889	3616.686	11	16840	21018.732
4000	12751.902	197634	10425.640	12	33767	21066.813
4500	No solution	No solution	14438.587	11	37842	21170.547
5000	No solution	No solution	No solution	No solution	No solution	No solution

\cap	
¥	
2	
<u> </u>	
uantity of 1	
Эf	
~	
5	
Ē	
E	
ñ	
th a supply q	
>	
Ξ.	
đ	
h	
\mathbf{v}	
а	
h	
ij	
≥	
e model wit	
<u>e</u>	
č	
ă	
H	
õ	
ũ	
Š	
a	
<u> </u>	
Š	
. E	
r ni	
,0	
Ŧ	
Ξ	
Ę.	
÷	
cut algorithm	
50	
al	
÷.	
n	
. <u>с</u>	
It	
nult	
mult	
d mult	
nd mult	
and mult	
er and mult	
ver and mult	
olver and mult	
solver and multicut algorithm for risk-averse model with a supply quantity of 1,0	
K solver and mult	
EX solver and mult	
EX solver and mult	
PLEX solver and mult	
CPLEX solver and mult	
CPLEX	
CPLEX	
n of CPLEX solver and mult	
CPLEX	
CPLEX	
ison of CPLEX	
ison of CPLEX	
ison of CPLEX	
CPLEX	
omparison of CPLEX	
ison of CPLEX	
omparison of CPLEX	
omparison of CPLEX	
0: Comparison of CPLEX	
0: Comparison of CPLEX	
0: Comparison of CPLEX	
0: Comparison of CPLEX	

Based on the results given in Table 6.9 and Table 6.10, we first analyze the results of each solution method within itself, and then compare them with each other.

MSDL cannot be solved by CPLEX solver for number of scenarios higher than 4,000. For both supply quantities, elapsed times and number of CPLEX iterations in MSDL increases if number of scenarios increase.

Observation 6.3.1 Figure 6.2 and Figure 6.3 show the elapsed time and number of CPLEX iteration for MSDL with different scenario numbers and supply quantities. While the number of CPLEX iterations linearly increase with the number of scenarios, increase in elapsed times is not linear. This trend in elapsed times is due to the fact that both number of iterations and elapsed time for each iteration increase with an increase in the number of scenarios. Furthermore, supply quantities are not effective on elapsed times and number of CPLEX iterations.

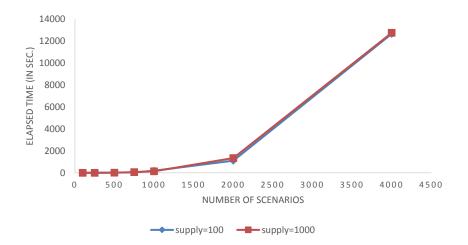


Figure 6.2: CPLEX elapsed time for MSDL

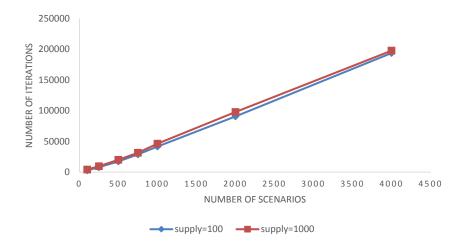


Figure 6.3: Number of CPLEX iterations for MSDL

Observation 6.3.2 Multicut algorithm for MSDL solves the test problems with a scenario number up to 4,500. Our test problems with number of scenarios higher than 4,500 can not be solved by the multicut algorithm for MSDL; which is an unexpected result for us.

Multicut algorithm for MSDL is proposed to solve the problems with high number of scenarios. In [37], Kristoffersen studies with two-stage stochastic linear programming model. We study with two-stage stochastic integer programming model. However, total unimodularity of our second-stage subproblem makes it possible for us to use the multicut algorithm method in [37]. Computational results of multicut algorithm for MSDL suggests that multicut algorithm method in [37] can be used for our case problem; nevertheless it is not effective to solve problems with high number of scenarios. Integrality restrictions in the first-stage problem (i.e. master problem in multicut algorithm) prevent the algorithm to be effective for the problems where more scenarios are involved.

Observation 6.3.3 For both supply quantities, the elapsed times of multicut algorithm for MSDL also raise with increasing number of scenarios as expected. Number of iterations do not have an increasing or decreasing trend depending on the scenario size. They are almost the same for each scenario size. However, number of cuts used in multicut algorithm increases with an increment in scenario size.

Observation 6.3.4 Preference between CPLEX solver and multicut algorithm de-

pends on the scenario size. While CPLEX is preferable for small sized problems, multicut algorithm can be more efficient for larger sized problems.

Figure 6.4 illustrates that CPLEX is more efficient to solve the test problems with up to 2,000 scenarios. However, at 4,000 scenarios, elapsed time of CPLEX is higher than the elapsed time of multicut algorithm. Moreover, CPLEX cannot solve the test problems with a scenario size more than 4,000. Thus, for a certain number of scenarios, multicut algorithm is preferable to CPLEX.

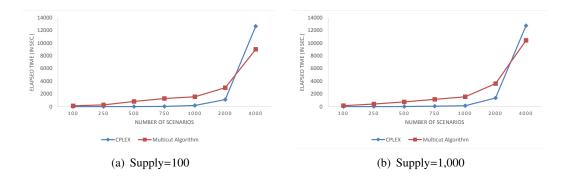


Figure 6.4: Comparison of elapsed times of CPLEX and multicut algorithm for MSDL

6.4 Comparison of the Risk-neutral and the Risk-averse Model

In this study, two different risk approaches are studied for the same problem. In the risk-averse approach, different than the risk-neutral problem formulation, expected excess from the expected cost over all scenarios is also minimized. Hence, in risk-averse approach, we expect scenario costs to have lower variances and expect the expected cost over all scenarios to be higher compared to the risk-neutral case. In order to understand the effect of risk factor in the problem environment, we run the risk-neutral and risk-averse models with the same design parameters. Models are run for 5 different problems with different number of scenarios and supply quantity 100 and 1,000. The results can be seen in Table 6.11 and Table 6.12. RN refers to the risk-neutral model and RA refers to the risk-averse model.

	To	tal Expecte	ed Cost	Standard Deviation of Scenario Costs			
Number of Scenarios	RN	RA	(RA-RN)/RN	RN	RA	(RA-RN)/RN	
100	1940.153	1941.565	0.07%	442.808	436.045	-1.53%	
250	1943.256	1946.504	0.17%	433.884	422.770	-2.56%	
500	1944.836	1945.901	0.05%	405.883	399.700	-1.52%	
750	1945.403	1950.173	0.25%	410.724	391.617	-4.65%	
1000	1927.651	1929.133	0.08%	410.894	402.388	-2.07%	

Table6.11: Comparison of the risk-neutral and the risk-averse models with a supply quantity 100

Table6.12: Comparison of the risk-neutral and the risk-averse models with a supply quantity 1,000

	Total Expected Cost				Standard Deviation of Scenario Costs			
Number of Scenarios	RN	RA	(RA-RN)/RN	RN	RA	(RA-RN)/RN		
100	19452.623	19504.508	0.27%	4242.404	4042.565	-4.71%		
250	19164.911	19190.132	0.13%	4357.700	4257.700	-2.29%		
500	19346.207	19371.275	0.13%	4412.421	4300.261	-2.54%		
750	19520.593	19539.109	0.09%	4186.300	4097.800	-2.11%		
1000	19033.914	19053.208	0.10%	4171.437	4084.570	-2.08%		

Table 6.11 and Table 6.12 show that standard deviation of scenario costs in risk-averse model is lower than the risk-neutral model. Total expected cost of risk-averse model is higher than the risk-neutral model. Results are compatible with our expectations.

Observation 6.4.1 Percent increase in total expected cost is lower than the percent decrease in standard deviations.

Observation 6.4.2 There is a relationship between the percent changes in total expected cost and in standard deviations. The more the increase in the percent change in total expected cost, the more the decrease in the percent change in standard deviations.

In the problem formulation, the first stage and the second stage unit transportation costs are assumed to be equal. Comparison of the risk-neutral and the risk-averse models are given with the same first stage and the second stage unit transportation costs in Table 6.11 and Table 6.12. In other words, if the first stage unit transportation costs are c, the second stage unit transportation costs are also c in Table 6.11 and Table 6.12. In other words, if the first stage unit transportation costs are c, the second stage unit transportation costs are also c in Table 6.11 and Table 6.12. In other words, if the first stage unit transportation costs are also c in Table 6.11 and Table 6.12. In other words, if the first stage unit transportation costs are also c in Table 6.11 and Table 6.12. In other words, if the first stage unit transportation costs are also c in Table 6.11 and Table 6.12. In other words, if the first stage unit transportation costs are also c in Table 6.11 and Table 6.12. In other words, if the first stage unit transportation costs are also c in Table 6.11 and Table 6.12. In other words, if the first stage unit transport to see the effect of different second stage unit costs in the comparison of risk-neutral and the risk-averse models, we double and triple the second stage unit

transportation costs while fixing the first stage costs. In other words, we keep the first stage unit transportation costs as c and change the second stage unit transportation costs to 2c and 3c. The results of the 2c and 3c second stage costs for supply quantities of 100 and 1,000 are given in Table A.1, Table A.2, Table A.3 and Table A.4 in Appendix A.

Observation 6.4.3 We compare the risk-neutral and the risk-averse models with c, 2c and 3c second stage unit costs in Table 6.13 and Table 6.14. It is seen that absolute change in total expected costs and absolute change in standard deviations are higher in the models with 2c second stage unit costs compared to models with c second stage unit costs. When the results of the models with 3c second stage unit costs are analyzed, Table 6.13 and Table 6.14 illustrate that absolute change in total expected costs and absolute change in standard deviations are higher in 3c compared to c. However, absolute changes in total expected cost and the standard deviations in the models with 3c second stage unit costs. Consequently, higher second stage costs than the first stage costs raise the absolute change in total expected cost and the standard deviations of the scenario costs compared to the case when the first and the second stage costs are equal. However, there is not any linear trend in the absolute changes of the total expected cost and the standard deviations of the scenario cost and the standard deviations with the increase in the second stage costs.

	Change	e in Total I (RA-RN	Expected Cost)/RN	Change in Standard Deviation of Scenario Costs (RA-RN)/RN		
Number of Scenarios	c	2c	3c	c	2c	3c
100	0.07%	0.63%	0.29%	-1.53%	-9.91%	-6.45%
250	0.17%	0.50%	0.38%	-2.56%	-8.13%	-6.72%
500	0.05%	0.67%	0.49%	-1.52%	-9.48%	-5.77%
750	0.25%	0.56%	0.50%	-4.65%	-8.49%	-8.75%
1000	0.08%	0.47%	0.41%	-2.07%	-7.10%	-5.77%

Table6.13: Comparison of the risk-neutral and the risk-averse models with different second stage unit costs for a supply quantity 100

Number of Scenarios	Change in Total Expected Cost (RA-RN)/RN			Change in Standard Deviation of Scenario Costs (RA-RN)/RN		
	c	2c	3c	c	2c	3c
100	0.27%	0.35%	0.42%	-4.71%	-8.89%	-6.80%
250	0.13%	0.71%	0.75%	-2.29%	-10.13%	-10.27%
500	0.13%	0.61%	0.57%	-2.54%	-9.56%	-8.98%
750	0.09%	0.42%	0.40%	-2.11%	-7.21%	-7.17%
1000	0.10%	0.48%	0.42%	-2.08%	-7.55%	-6.66%

Table6.14: Comparison of the risk-neutral and the risk-averse models with different second stage unit costs for a supply quantity 1,000

CHAPTER 7

CONCLUSION

In this study, initial inventory positioning problem of an auction company serving in floricultural industry is studied. The aim is to optimally position the products at auction centers prior to the clock minimizing the cost with the involved uncertainties about the buyers and their purchase quantities. Two different risk approaches are followed during the study.

First, case problem is studied in a risk-neutral environment. Two-stage stochastic integer programming is used to model the problem. The minimum cost flow formulation is used in the first and the second stages. In the literature, most of the methods used to solve two-stage stochastic integer programming are based on L-shaped decomposition. To handle a model with high number of scenarios, we also suggest L-shaped and multicut L-shaped decomposition methods. In L-shaped and multicut L-shaped methods, simplex multipliers of the optimal solution of the second stage subproblem are used to derive optimality cuts. In our study, total unimodularity of the second-stage subproblem of the case problem is utilized to obtain optimality cuts in the decomposition methods.

Second, risk-averse approach is applied to the problem. Risk-averse preferences in stochastic programming have drawn more attention lately. Mean risk models are a way to incorporate risk in optimization problems. We prefer first-order mean-semideviation as the risk measure. The reason beyond the choice of first-order mean-semideviation is its advantages in optimization problems. Objective function of the risk-averse formulation is nonlinear and nondifferentiable. We linearize it and propose a decomposition based solution algorithm, which resembles multicut L-shaped

method, for the linearized formulation.

During the computational study, we observe that optimal solution of the LP relaxation of RN2SSIP is the same as RN2SSIP. Moreover, it is seen that CPLEX can solve high number of scenarios to a certain degree. L-shaped and multicut L-shaped decomposition algorithms are not computationally efficient for the test problems that CPLEX can solve. They may be useful only for the problems with high number of scenarios that CPLEX cannot. When we compare the two decomposition methods, it can be said that multicut decomposition algorithm is more efficient than L-shaped method for the test problems considering the computational times. Furthermore, solving the stochastic problem instead of using the expected value solution as the first stage decision brings about expected cost decrease.

Looking at the computational results for the risk-averse model, it can be said that while CPLEX is preferable for small sized problems, the multicut decomposition algorithm is more efficient than CPLEX for larger sized problems.

In addition, we also compare the risk-averse and the risk-neutral models. In the riskaverse case, expected cost of the scenarios increase while variation of scenario costs decrease compared to the risk-neutral model as expected. If the second stage unit transportation costs are taken higher than the first stage unit transportation costs, the absolute changes in expected cost and the variations of the scenario costs increase.

In this thesis, in the risk-averse context, semideviation is used as the risk measure. Nevertheless, CVaR is also a preferable risk measure in optimization problems. As a future study, CVaR can be used as a risk measure in the problem context. Moreover, the multicut algorithm suggested for problems with high number of scenarios is not effective as we expect. The model may require further extensions as Branch and Cut or alternative methods. The multicut algorithm can be improved as a future study. In addition, the problem is formulated and solved for one grower, it can be extended to a multigrower case.

REFERENCES

- [1] Birge, J. R., & Louveaux, F. (1997). *Introduction to stochastic programming*. (pp. 83-262). New York : Springer.
- [2] Dantzig, G. B. (1955). Linear programming under uncertainty. *Management Science*, 1(3-4), 197-206.
- [3] Beale, E. M. L. (1955). On minimizing a convex function subject to linear inequalities. J. Royal Statistical Society. Series B (Methodological), 17(2), 173-184.
- [4] Van Slyke, R. M., & Wets, R. (1969). L-shaped linear programs with application to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4), 638-663.
- [5] Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1), 238–252.
- [6] Neise, F. (2008). *Risk management in stochastic integer programming with application to dispersed power generation* (pp. 1-26). Wiesbaden: Vieweg Teubner.
- [7] Ahmed, S. (2011). Two-stage stochastic integer programming: A brief introduction. *Wiley Encyclopedia of Operations Research and Management Science*
- [8] Carøe, C. C, & Schultz, R. (1999). Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24, 37-45.
- [9] Wollmer, R. D. (1980). Two stage linear programming under uncertainty with 0-1 integer first stage variables. *Mathematical Programming*, 19, 279-288.
- [10] Laporte, G., & Louveaux, F. V. (1993). The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13, 133-142.
- [11] Zheng, Q. P., Wang, J., Pardalos, P. M., & Guan, Y. (2013). A decomposition approach to the two stage stochastic unit commitment problem. *Ann Oper Res*, 210, 387-410.
- [12] Escudero, L. F., Garín M.A., Pérez, G., & Unzueta, A. (2013). Computers & Operations Research, 40, 362-377.
- [13] Ntaimo, L. (2013). Fenchel decomposition for stochastic mixed-integer programming. J Glob Optim, 55, 141-163.
- [14] Gade, D., Küçükyavuz, S., & Sen, S. (2014). Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs. *Math. Pro*gram., A(144), 39-64.

- [15] Carøe C. C., & Tind, J. (1998). L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming*, 83, 451-464.
- [16] Ahmed, S., Tawarmalani, M., & Sahinidis, N. V. (2004). A finite branch-andbound algorithm for two-stage stochastic integer programs. *Math. Program.*, A(100), 355-377.
- [17] Hemmecke, R., & Schultz, R. (2003). Decomposition of test sets in stochastic integer programming. *Math. Program.*, B(94), 323-341.
- [18] Kong N., Schaefer, A. J., & Hunsaker, B. (2006). Two-stage integer programs with stochastic right-hand sides: a superadditive dual approach. *Math. Pro*gram., B(108), 275-296.
- [19] Sherali, H. D., & Fraticelli B. M. P. (2002). A modification of Bender's decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. *Journal of Global Optimization*, 22, 319-342.
- [20] Sen, S., & Higle, J. L. (2005). The C³ Theorem and a D² algorithm for large scale stochastic mixed-integer programming: Set convexification. *Math. Program.*, A(104), 1-20.
- [21] Sen, S., & Sherali, H. D. (2006). Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Math. Program.*, A(106), 203-223.
- [22] Shapiro, A., Dentcheva, D., & Ruszczyński, A. (2009). Lectures on stochastic programming: Modeling and theory (pp. 253-260). Philadelphia: Society for Industrial and Applied Mathematics.
- [23] Markowitz, H. M. (1952). Portfolioselection. J. Finance, 7, 77–91.
- [24] Artzner, P., Delbean, F., Eber, J. M., & Heath, D. (1999). Coherent measures of risk. *Mathematical Finance*, 9(3), 203-228.
- [25] Miller, N., & Ruszczyński, A. (2011). Risk-averse two-stage stochastic linear programming: Modelling and decomposition. *Operations Research*, 59(1), 125-132.
- [26] Krishna, V. (2002). Auction Theory. (p.2.). San Diego: Academic Press.
- [27] Huang, D., Zhu, S. S., Fabozzi, F. J., & Fukushima, M. (2008). Portfolio selection with uncertain exit time: A robust CVaR approach. *Journal of Economic Dynamics & Control*, 32, 594-623.
- [28] Yao, H., Li, Z., & Lai, Y. (2013). Mean-CVaR portfolio selection: A nonparametric estimation framework. *Computers & Operations Research*, 40, 1014-122.
- [29] Dai, C., Cai, X.H., Cai, Y.P., Huo, Q, Lv, Y., & Huang, G. H. (2014). An interval-parameter mean-CVaR two-stage stochastic programming approach for waste management under uncertainty. *Stoch Environ Res Risk Assess*, 28, 167-187.

- [30] Gebreslassie, B. H., Yoa, Y., & You, F. (2012). Design under uncertainty of hydrocarbon biorefinery supply chains: Multiobjective stochastic programming models, decomposition algorithm, and a comparison between CVar and downside risk. *AIChE Journal*, 58 (7), 2155-2179.
- [31] Noyan, N. (2012). Risk-averse two-stage stochastic programming with an application to disaster management. *Computers & Operations Research*, 39, 541-559.
- [32] Soleimani, H., & Govindan, K. (2014). Reverse logistics network design and planning utilizing conditional value at risk. *European Journal of Operational Research*, 237, 487-497.
- [33] Xie, Y.L. & Huang, G. H. (2014). An optimization model for water resources allocation risk analysis under uncertainty. *Journal of Hydroinformatics*, 16(1), 144-164.
- [34] Zheng, Q. P., & Pardalos, P. M. (2010). Stochastic and risk management models and solution algorithm for natural gas transmission network expansion and LNG terminal location planning. *J Optim Theory Appl*, 147, 337-357.
- [35] Ahmed, S. (2006). Convexity and decomposition of mean-risk stochastic programs. *Math. Program.*, A(106), 433-446.
- [36] Ogryczak, W., & Ruszczyński, A. (1999). From stochastic dominance to meanrisk models: Semideviations as risk measures. *European Journal of Operational Research*, 116, 33-50.
- [37] Kristoffersen, T. K. (2005). Deviation measures in linear two-stage stochastic programming. *Math. Math. Oper. Res.*, 62, 255-274.
- [38] Liu, C., Fan, Y., & Ordónez, F. (2009). A two-stage stochastic programming model for transportation network protection. *Computers & Operations Research*, 36, 1582-1590.
- [39] Carøe, C. C. & Tind, J. (1997). A cutting-plane approach to mixed 0–1 stochastic integer programs. *European Journal of Operational Research*, 101 (2), 306-316.
- [40] Ntaimo, L., & Sen, S. (2005). A branch-and-cut algorithm for two-stage stochastic mixed-integer programs with continuous first-stage variables. *International Journal of Computational Science and Engineering*, 3(6), 232–241.
- [41] Märkert, A., & Schultz, R. (2005). On deviation measures in stochastic integer programming. *Operations Research Letters*, 33(5), 441-449.
- [42] FloraHolland. Facts & Figures. https://www.floraholland.com/media/2460310/ Kengetallen-EN-2013.pdf. Retrieved: 1 July 2014.
- [43] Jonkman, P. (2010). Investigating the feasibility and desirability of a coordinated organization of inter-auction transport in the floricultural supply chain. Master's thesis, Erasmus University Rotterdam, The Netherlands.
- [44] Gaki, M. (2012). Coordination and optimization of inventory locations and flows in the FloraHolland Network. Master's thesis, Eindhoven University of Technology, The Netherlands.

- [45] Dat, D. (2010). Coordination and optimization of the inter-auction transport of floricultural products at FloraHolland. Master's thesis, Eindhoven University of Technology, The Netherlands.
- [46] Kong, N., Schaefer, A. J., & Ahmed, S. (2013). Totally unimodular stochastic programs. *Math. Program.*, 138(1-2), 1-13.
- [47] Paterson, C., Kiesmüller, G., Teunter, R., & Glazebrook, K. (2011). Inventory models with lateral transshipments: A review. *European Journal of Operational Research*, 210, 125-136.
- [48] Birge, J. R., & Louveaux, F. V. (1988). A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operations Research*, 34, 384–392.

APPENDIX A

COMPARISON OF THE RISK-NEUTRAL AND THE RISK-AVERSE MODEL WITH 2C AND 3C SECOND STAGE UNIT TRANSPORTATION COSTS

TableA.1: Comparison of the risk-neutral and the risk-averse models with 2c second stage unit transportation costs for a supply quantity of 100

	Total Expected Cost			Standard Deviation of Scenario Costs		
Number of Scenarios	RN	RA	(RA-RN)/RN	RN	RA	(RA-RN)/RN
100	2773.075	2790.535	0.63%	759.598	684.305	-9.91%
250	2759.242	2772.951	0.50%	715.8365	657.668	-8.13%
500	2722.420	2740.547	0.67%	696.594	630.545	-9.48%
750	2738.079	2753.440	0.56%	661.613	605.420	-8.49%
1000	2710.373	2723.140	0.47%	688.590	639.6722	-7.10%

TableA.2: Comparison of the risk-neutral and the risk-averse models with 3c second stage unit transportation costs for a supply quantity of 100

	Total Expected Cost			Standard Deviation of Scenario Costs		
Number of Scenarios	RN	RA	(RA-RN)/RN	RN	RA	(RA-RN)/RN
100	3517.418	3527.450	0.29%	1026.500	960.295	-6.45%
250	3476.795	3489.861	0.38%	988.4554	922.033	-6.72%
500	3415.636	3432.238	0.49%	944.725	890.238	-5.77%
750	3438.582	3455.839	0.50%	923.376	842.589	-8.75%
1000	3400.118	3413.915	0.41%	957.250	902.0342	-5.77%

TableA.3: Comparison of the risk-neutral and the risk-averse models with 2c second stage unit transportation costs for a supply quantity of 1,000

	Total Expected Cost			Standard Deviation of Scenario Costs		
Number of Scenarios	RN	RA	(RA-RN)/RN	RN	RA	(RA-RN)/RN
100	27499.784	27595.376	0.35%	6826.500	6219.300	-8.89%
250	27148.615	27341.31	0.71%	7264.200	6528.500	-10.13%
500	27273.531	27439.916	0.61%	7328.100	6627.400	-9.56%
750	27450.001	27564.600	0.42%	6929.400	6429.800	-7.21%
1000	26887.571	27015.826	0.48%	6961.800	6436.100	-7.55%

	Total Expected Cost				Standard Deviation of Scenario Costs		
Number of Scenarios	RN	RA	(RA-RN)/RN	RN	RA	(RA-RN)/RN	
100	34364.128	34506.902	0.42%	9105.100	8486.000	-6.80%	
250	34262.14	34519.48	0.75%	9943.500	8922.600	-10.27%	
500	34320.760	34515.112	0.57%	10067.000	9162.800	-8.98%	
750	34443.96	34582.133	0.40%	9560.400	8875.000	-7.17%	
1000	33834.069	33977.127	0.42%	9601.300	8961.800	-6.66%	

TableA.4: Comparison of the risk-neutral and the risk-averse models with 3c second stage unit transportation costs for a supply quantity of 1,000